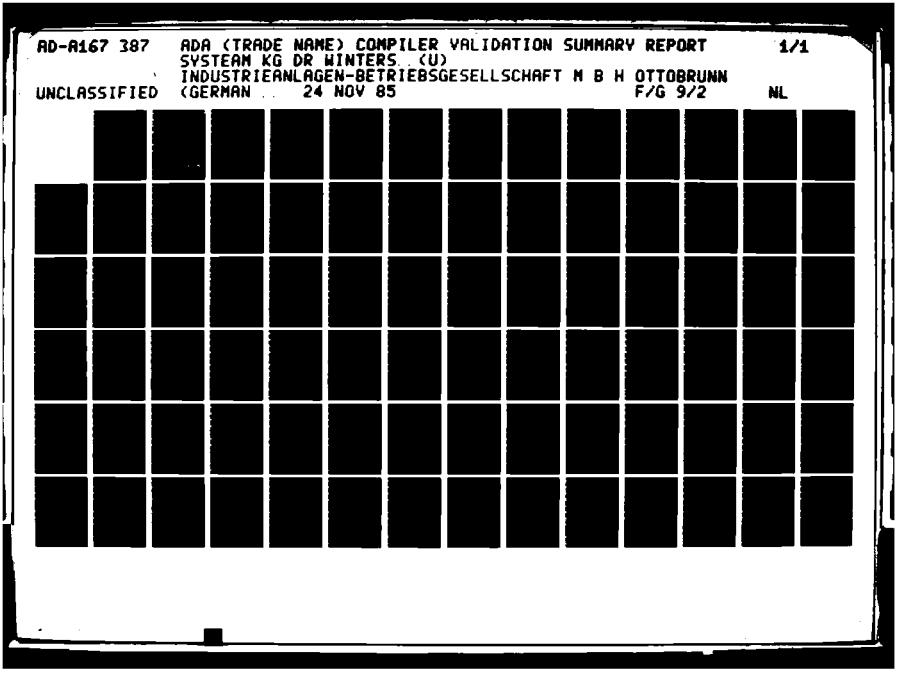
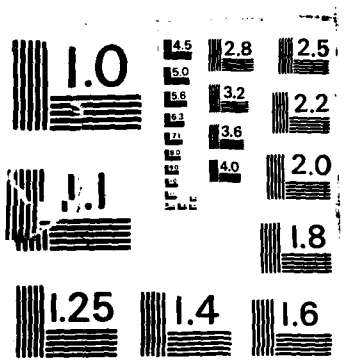

AD-A167 387 ADA (TRADE NAME) COMPILER VALIDATION SUMMARY REPORT 1/1
SYSTEAM KG DR WINTERS (U)
INDUSTRIEANLAGEN-BETRIEBSGESELLSCHAFT M B H OTTOBRUNN
UNCLASSIFIED (GERMAN 24 NOV 85 F/G 9/2 NL



AD-A167 387

REPORT DOCUMENTATION PAGE.	READ INSTRUCTIONS BEFORE COMPLETING FORM
1 REPORT NUMBER / 2 GOVT ACCESSION NO.	3 RECIPIENT'S CATALOG NUMBER
4 TITLE (and Subtitle) Ada* Compiler Validation Summary Report: Systeam KG. Dr. Winterstein, Systeam/German MoD VAX-11 Compiler, VAX-11/750	5 TYPE OF REPORT & PERIOD COVERED 24 Nov 1985-24 Nov 1986 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Industrieanlagen-Betriebsgesellschaft m.b.H (IABG)	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Industrieanlagen-Betriebsgesellschaft m.b.H (IABG) Detp SZT, Einsteinstrasse, D 8012 Ottobrunn Germany	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office 1211 S. Fern St., Room C-107 Arlington, VA 22202	12. REPORT DATE 24 Nov 1985 13 NUMBER OF PAGES 81
14 MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Industrieanlagen-Betriebsgesellschaft m.b.H. (IABG) Dept SZT , Einsteinstrasse D 8012 Ottobrunn, Germany	15. SECURITY CLASS. (of this report) unclassified 15a DECLASSIFICATION/DOWNGRADING SCHEDULE

16 DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

unclassified

18. SUPPLEMENTARY NOTES

*Ada is a registered trademark of the U.S. Government (Ada Joint Program Office)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ada Programming Language, Ada Compiler Validation Summary Report, Ada Compiler Validation Capability, ACVC, Validation Testing, Ada Validation Office, AVO, Ada Validation Facility, AVF, ANSI/MIL-STD-1815A, Ada Joint Program Office, AJPO

20. ABSTRACT (Continue on reverse side if necessary and identify by block number).

see attached abstract

DTIC FILE COPY

DTIC ELECTE MAY 0 1 1986 S D E

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Ada* COMPILER VALIDATION SUMMARY REPORT:
Systeam KG. Dr. Winterstein
Systeam/German MoD VAX-11 Compiler
VAX-11/750

Completion of On-Site Validation:
85-11-24

Prepared By:
Industrieanlagen-Betriebsgesellschaft m.b.H. (IABG)
Dept SZT
Einsteinstrasse
D 8012 Ottobrunn

Prepared For:
Ada Joint Program Office
United States Department of Defense
Washington, D.C.

---

* Ada is a registered trademark of the United States Government (Ada Joint Program Office)

Ada* Compiler Validation Summary Report:

Compiler Name: Systeam/German MoD VAX-11 Compiler

Host and Target Computer

VAX-11/750 under VMS 4.1

Testing Completed 85-11-24 Using ACVC 1.6

This report has been reviewed and approved:

IABG m.b.H.
Dr. H. Hummel
Dept SZT
Einsteinstrasse
D 8012 Ottobrunn

Ada Validation Office (AVO)
Thomas H. Probert, Ph.D.
Institute for Defense Analyses
Alexandria, VA

Ada Joint Program Office (AJPO)
Virginia L. Castor
Director
Washington, D.C.

* Ada is a registered trademark of the United States Government (Ada Joint Program Office)

EXECUTIVE SUMMARY

This Validation Summary Report presents the results and conclusions of testing performed on the Systeam/German MoD VAX-11 Compiler. Standardized tests serve as input to an Ada compiler, producing results which are evaluated by the validation team. This summary briefly states the highlights of the Systeam/German MoD VAX-11 Compiler validation.

On-site testing was performed 85-11-22 through 85-11-24 at Karlsruhe under the auspices of the IABG m.b.H. (AVF), according to Ada Validation Office policies and procedures. The Systeam/German MoD VAX-11 Compiler is hosted on VAX-11/750 operating under VMS 4.1. The suite of tests known as the Ada Compiler Validation Capability (ACVC), Version 1.6, was used. The ACVC is used to validate conformance of a compiler to ANSI/MIL-STD-1815A Ada. The purpose of testing is to ensure that a compiler properly implements legal language constructs and that it identifies and rejects illegal language constructs. The testing also identifies behavior that is implementation dependent but permitted by the Ada Standard. Six classes of tests are used. These tests are designed to perform checks at compile time, at link time, or during execution.

The results of validation are summarized in the following table.

RESULT	TEST CLASS						TOTAL
	A	B	C	D	E	L	
Passed	61	769	976	17	8	1	1832
Failed	0	0	0	0	0	0	0
Inapplicable	0	12	250	0	0	2	264
Anomalous	0	0	0	0	0	0	0
Withdrawn	0	19	47	0	0	0	66
TOTAL	61	800	1273	17	8	3	2162

Tests found to contain errors were withdrawn from Version 1.6 of the Ada Compiler Validation Capability (ACVC).

Some tests demonstrate that language features are not supported by an implementation. For this implementation the tests determined the following.

- SHORT_INTEGER is not supported:
  B52004E-AB.DEP B55B09D-AB.DEP B86001CR-AB.DEP

C34001D-B.DEP C55B07B-AB.DEP

. LONG_INTEGER is not supported:
B52004D-AB.DEP B55B09C-AB.DEP B86001CS-AB.DEP
C34001E-B.DEP C55B07A-AB.DEP
. SHORT_FLOAT is not supported:
B86001CP-AB.DEP C34001F-B.DEP C35702A-AB.DEP

. LONG_FLOAT is not supported:

B86001CQ-AB.DEP C34001G-B.DEP C35702B-AB.DEP

. No other integer type other than INTEGER, SHORT_INTEGER, AND LONG_INTEGER is supported:

B86001DT-AB.TST

. The package SYSTEM is used by package TEXT_IO:

C86001F-B.DEP

. Pragma INLINE is not supported for procedures:

LA3004A-AB.ADA

. Pragma INLINE is not supported for functions:

LA3004B-AB.ADA

. Mode IN_FILE is supported (for sequential I/O):

CE2102D-B.ADA

. Mode OUT_FILE is supported (for sequential I/O):

CE2102E-B.ADA

. Mode INOUT_FILE is supported (for direct I/O):

CE2102F-B.ADA

. Mode RESET and DELETE are supported (for sequential and direct I/O):

CE2102G-B.ADA

. No more than one internal file can be associated with the same external file except for mode IN_FILE:

CE2107B-B.ADA CE2107C-B.ADA
CE2107D-B.ADA CE2111D-B.ADA
CE3111B-B.ADA CE3111C-B.ADA
CE3114B-B.ADA

. Instantiation of package DIRECT_IO with unconstrained array types is supported only if a form parameter is used. Therefore

CE2401D-B.DEP

was modified.

. An external file associated with more than one internal file can be reset for mode IN_FILE only:

CE3115A-B.ADA

ACVC Version 1.6 was taken on-site via magnetic tape to Karlsruhe. The tape was loaded, and all tests, except for the executable tests which make use of a floating point precision greater than SYSTEM.MAX_DIGITS, were compiled on VAX-11/750. Class A, C, D, and E tests were executed on VAX-11/750.

On completion of testing, all results were analyzed for failed Class A, C, D, or E programs, and all Class B and L compilation results were individually analyzed.

The ACVC, Version 1.6, contains 2198 tests of which 1868 were applicable to Systeam/German MoD VAX-11 Compiler. No anomalies were found in the testing of this compiler. Testing demonstrated that all applicable tests were passed by this compiler and conformed to the Ada Standard. The AVF concluded that the results show acceptable compliance to ANSI/MIL-STD-1815A Ada.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The Validation Summary Report describes how an Ada compiler conforms to the language standard. This report explains all technical terms used within and thoroughly reports the Ada Compiler Validation Capability (ACVC) test results. Ada compilers must be written according to the language specification as given in the Ada Standard, ANSI/MIL-STD-1815A. All implementation-defined features must be included for the compiler to conform to the Standard. Following the guidelines of the Standard ensures continuity between compilers. That is, the entire Standard must be implemented, and nothing can be implemented that is not in the Standard.

Even though all validated Ada compilers conform to the Standard, it must be understood that some differences do exist between implementations. The Standard permits some implementation dependencies, e.g., the maximum length of identifiers, the maximum values of integer types, etc.. These implementation-dependent features limit the portability of programs between compilers. Other differences between compilers are due to limitations imposed on a compiler by the operating system and by the hardware. All of these dependencies are given in the report.

Validation summary reports are written according to a standardized format. Compiler users can, therefore, more easily compare the reports from several compilers when selecting a compiler for a given task. The validation report can be completed mostly from the test results produced during validation testing. Additional testing information is given at the end of the report and states problems and details which are unique for a specific compiler. The format of the validation report limits variance between reports, enhances readability of the report, and accelerates report readiness.

1 INTRODUCTION

1.1 Purpose of this Validation Summary Report

The Validation Summary Report documents the results of the testing performed on an Ada compiler. Testing was carried out for the following purposes:

- To identify any language constructs supported by the translator that do not conform to the Ada Standard

- To identify any unsupported language constructs required by the Ada Standard
- To describe the implementation-dependent behavior allowed by the Ada Standard

Testing of this compiler was conducted by IABG m.b.H. according to policies and procedures established by the Ada Validation Office (AVO). Testing was conducted from 85-11-Z2 through 85-11-24 at Karlsruhe.

1.2 Use of this Validation Summary Report

Consistent with the national laws of the originating country, the Ada Validation Office may make full and free public disclosure of this report. In the United States, this is provided in accordance with the "Freedom of Information Act" (5 U.S.C. /552). The results of this validation apply only to the computers, operating systems, and compiler versions identified in this report.

The organizations represented on the signature page of this report do not represent or warrant that any statement or statements set forth in this report are accurate or complete, or that the subject compiler has no nonconformances to the Ada Standard other than those presented. This report is not intended for the purpose of publicizing the findings summarized herein.

Questions regarding this report or the validation tests should be directed to:

Ada Validation Office
Institute for Defense Analyses
1801 N. Beauregard
Alexandria VA 22311

and to:

IABG m.b.H., Dept SZT
Einsteinstrasse
D 8012 Ottobrunn

1.3 REFERENCES

1. Reference Manual for the Ada Programming Language, ANSI/MIL-STD-1815A,

2. Ada Validation Organization: Policies & Procedures, MITRE Corporation, Jun 1982.

3. Compiler Validation Capability Implementers' Guide, SofTech, Inc., Dec 1984.

1.4 DEFINITION OF TERMS

Anomaly	A test result that, given pre-validation analysis, is not expected during formal validation but is judged allowable under the circumstances.
ACVC	The Ada Compiler Validation Capability. A set of programs that evaluates the conformance of a compiler to the Ada language specification, ANSI/MIL-STD-1815A.
Ada Standard	ANSI/MIL-STD-1815A, February 1983.
Applicant	The agency requesting validation.
AVF	The IABG m.b.H. In the context of this report, the AVF is responsible for conducting compiler validations according to established policies and procedures.
AVO	The Ada Validation Office. In the context of this report, the AVO is responsible for setting policies and procedures for compiler validations.
Compiler	A processor for the Ada language. In the context of this report, a compiler is any language processor, including cross-compilers, translators, and interpreters.
Failed test	A test for which the compiler generates a result that demonstrates nonconformance to the Ada Standard.
Host	The computer on which the compiler resides.
Inapplicable test	A test that uses features of the language that a compiler is not required to support or may legitimately support in a way other than the one expected by the test.
Passed test	A test for which a compiler generates the expected result.
Target	The computer for which a compiler generates code.
Test	A program that evaluates the conformance of a compiler to a language specification. In the context of this report, the term is used to designate a single ACVC test. The text of a program may be the text of one or more compilations.

Withdrawn test	A test that has an invalid test objective, fails to meet its test objective, or contains illegal use of the language.

1.5 Configuration

The candidate compilation system for this validation was tested under the configuration:

Compiler: Systeam/German MoD VAX-11 Compiler

Test Suite: Ada Compiler Validation Capability, Version 1.6

Host Computer:

Machine(s):	VAX-11/750
Operating System:	VMS 4.1
Memory Size:	8 MB
Disk System:	3xSI9751, 1xSI9784

Target Computer:

Machine(s):	VAX-11/750
Operating System:	VMS 4.1
Memory Size:	8 MB
Disk System:	3xSI9751, 1xSI9784

CHAPTER 2

TEST RESULTS

2.1 ACVC Test Classes

Conformance to ANSI/MIL-STD-1815A is measured using the Ada Compiler Validation Capability (ACVC). The ACVC contains both legal and illegal Ada programs structured into six test classes: A, B, C, D, E, and L. Legal programs are compiled and executed while illegal programs are just compiled. Support packages are used to report the results of the legal programs. A compiler must correctly process each of the tests in the suite and demonstrate conformance to the Ada Standard by either meeting the pass criteria given for the test or by showing that the test is inapplicable to the implementation. Tests that are found to contain errors are withdrawn from the ACVC. Detailed test results are listed in the Appendix D. The results of validation testing are summarized in the following table:

RESULT	TEST CLASS						TOTAL
	A	B	C	D	E	L	
Passed	61	769	976	17	8	1	1832
Failed	0	0	0	0	0	0	0
Inapplicable	0	12	250	0	0	2	264
Anomalous	0	0	0	0	0	0	0
Withdrawn	0	19	47	0	0	0	66
TOTAL	61	800	1273	17	8	3	2162

A total of 1868 tests were processed during this validation attempt. The 66 withdrawn tests in Version 1.6 were not processed, nor were 250 Class C tests that were inapplicable mostly because they use floating point types having digits that exceed the maximum value for the implementation. All other tests were processed.

Some conventions are followed in the ACVC to ensure that the tests are reasonably portable without modification. For example, the tests make use of only the basic 55 character set, contain lines with a maximum length of 72 characters, use small numeric values, and place features that may not be supported in separate tests. However, some tests contain values that require the test to be customized according to implementation-specific values. The values used for this validation are listed in Appendix B.

2.1.1 Class A Tests

Class A tests check that legal Ada programs can be successfully compiled and executed. However, no checks are performed during execution to see if the test objective has been met. For example, a Class A test checks that reserved words of another language (other than those already reserved in the Ada language) are not treated as reserved words by an Ada compiler. A Class A test is passed if no errors are detected at compile time and the program executes to produce a message indicating that it has passed. If a Class A test cannot be compiled and executed because of its size, then the test is split into a set of smaller subtests that can be processed. No splits were required.

The following table shows that all applicable Class A tests were passed:

RESULT	CHAPTER												
	2	3	4	5	6	7	8	9	10	11	12	14	TOTAL
Passed	13	6	0	5	2	12	13	3	0	0	0	7	61
Failed	0	0	0	0	0	0	0	0	0	0	0	0	0
Inapplicable	0	0	0	0	0	0	0	0	0	0	0	0	0
Anomalous	0	0	0	0	0	0	0	0	0	0	0	0	0
Withdrawn	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	13	6	0	5	2	12	13	3	0	0	0	7	61

2.1.2 Class B Tests

Class B tests check that a compiler detects illegal language usage. Class B tests are not executable. Each test in this class is compiled and the resulting compilation listing is examined manually to verify that every syntax or semantic error in the test is detected. A Class B test is passed if every illegal construct that it contains is detected by the compiler. If one or more errors are not detected, then a version of the test is created that contains only the undetected errors. The resulting "split" is compiled and examined. The splitting process continues until all errors are detected by the compiler. Splits were required for 4 tests:

B37301A B950ABA BC10AE1 BC10AEB

The following table shows that all applicable Class B tests were passed:

RESULT	CHAPTER												
	2	3	4	5	6	7	8	9	10	11	12	14	TOTAL
Passed	35	72	83	107	70	55	46	92	35	8	148	18	769
Failed	0	0	0	0	0	0	0	0	0	0	0	0	0
Inapplicable	0	0	0	6	0	0	6	0	0	0	0	0	12
Anomalous	0	0	0	0	0	0	0	0	0	0	0	0	0
Withdrawn	0	1	0	0	3	2	0	0	1	0	12	0	19
TOTAL	35	73	83	113	73	57	52	92	36	8	160	18	800

2.1.3 Class C Tests

Class C tests check that legal Ada programs can be correctly compiled and executed. Each Class C test is self-checking and produces a PASS/FAIL message indicating the result when it is executed. If a Class C test cannot be compiled because it exceeds the compiler's capacity, then the test is split into smaller subtests until all are compiled and executed. No splits were required.

The following table shows that all applicable Class C tests were passed:

RESULT	CHAPTER												
	2	3	4	5	6	7	8	9	10	11	12	14	TOTAL
Passed	20	101	168	119	70	14	96	104	36	20	55	171	976
Failed	0	0	0	0	0	0	0	0	0	0	0	0	0
Inapplicable	22	106	101	0	0	0	1	1	0	0	0	19	250
Anomalous	0	0	0	0	0	0	0	0	0	0	0	0	0
Withdrawn	0	1	27	0	4	0	0	4	7	0	0	4	47
TOTAL	42	208	296	119	74	14	97	109	43	20	55	196	1273

2.1.4 Class D Tests

Class D tests check the compilation and execution capacities of a compiler. Since there are no requirements placed on a compiler by the Ada Standard for the number of identifiers permitted in a compilation, the number of units in a library, the number of nested loops in a subprogram body, and so on, a compiler may refuse to compile a Class D test. Each Class D test is self-checking and produces a PASS/FAIL message indicating the result when it is executed. If a Class D test fails to compile because the capacity of the compiler is exceeded, then the test is classified as inapplicable.

The following table shows that all applicable Class D tests were passed:

RESULT	CHAPTER												
	2	3	4	5	6	7	8	9	10	11	12	14	TOTAL
Passed	1	0	4	9	3	0	0	0	0	0	0	0	17
Failed	0	0	0	0	0	0	0	0	0	0	0	0	0
Inapplicable	0	0	0	0	0	0	0	0	0	0	0	0	0
Anomalous	0	0	0	0	0	0	0	0	0	0	0	0	0
Withdrawn	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	1	0	4	9	3	0	0	0	0	0	0	0	17

Capacities measured by the Class D tests are detailed in section 2.4, IMPLEMENTATION CHARACTERISTICS.

2.1.5 Class E Tests

Class E tests provide information about the compiler in those areas in which the Ada Standard permits implementations to differ. Each Class E test is executable and produces messages that indicate how the Ada Standard is interpreted. However, in some cases the Ada Standard permits a compiler to detect a condition either at compile time or at execution time, and thus a Class E test may correctly fail to execute. A Class E test is passed if it fails to compile and appropriate error messages are issued, or if it executes properly and produces a message that it has passed. If a Class E test cannot be compiled and executed because of its size, then the test is split into a set of smaller subtests that can be processed. No splits were required.

The following table shows that all applicable Class E tests were passed:

RESULT	CHAPTER												
	2	3	4	5	6	7	8	9	10	11	12	14	TOTAL
Passed	1	3	2	1	0	0	0	0	0	0	0	1	8
Failed	0	0	0	0	0	0	0	0	0	0	0	0	0
Inapplicable	0	0	0	0	0	0	0	0	0	0	0	0	0
Anomalous	0	0	0	0	0	0	0	0	0	0	0	0	0
Withdrawn	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	1	3	2	1	0	0	0	0	0	0	0	1	8

Information obtained from the Class E tests is detailed in section 2.4, IMPLEMENTATION CHARACTERISTICS.

2.1.6 Class L Tests

Class L tests check that incomplete or illegal Ada programs involving multiple, separately compiled units are detected and not allowed to execute. Class L tests are compiled separately and execution is attempted. A Class L test passes if it is rejected at link time and the test does not execute.

The following table shows that all applicable Class L tests were passed:

RESULT	CHAPTER												
	2	3	4	5	6	7	8	9	10	11	12	14	TOTAL
Passed	0	0	0	0	0	0	0	0	1	0	0	0	1
Failed	0	0	0	0	0	0	0	0	0	0	0	0	0
Inapplicable	0	0	0	0	0	0	0	0	2	0	0	0	2
Anomalous	0	0	0	0	0	0	0	0	0	0	0	0	0
Withdrawn	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	0	0	0	0	0	0	0	0	3	0	0	0	3

2.1.7 Support Units

Three packages support the self-checking features of Class C tests: REPORT, CHECK_FILE, and VAR_STRINGS. The REPORT package provides the mechanism by which executable tests report results. It also provides a set of identity functions that are used to defeat some compiler optimization strategies to cause computations to be made by the target computer instead of the by the compiler on the host computer. The CHECK_FILE package is used to check the contents of text files written by some of the Class C tests for Chapter 14 of the Ada Standard. The VAR_STRINGS package defines types and subprograms for manipulating varying-length character strings. The operation of these three packages is checked by a set of executable tests. These tests produce messages that are examined manually to verify that the packages are operating correctly. If these packages are not operating correctly, then validation is not attempted.

An applicant is permitted to substitute the body of package REPORT with an equivalent one if for some reason the original version provided by the ACVC cannot be executed on the target computer. Package REPORT was modified for this validation in order to print the date and time of execution.

All support package specifications and bodies were compiled and were demonstrated to be operating correctly.

2.2 Withdrawn Tests

Some tests are withdrawn from the ACVC because they do not conform to the Ada Standard. When testing was performed, 66 tests had been withdrawn for the reasons indicated in Appendix E.

2.3 Description of Inapplicable Tests

201 tests were not processed because SYSTEM.MAX_DIGITS is 9. These tests were:

		C45321F,G,...,Y-B
C35705F,G,...,Y-B	C35708F,G,...,Y-B	C45421F,G,...,Y-B
C35706F,G,...,Y-B	C35802F,G,...,Y-B	C45424F,G,...,Y-B
C35707F,G,...,Y-B	C45241F,G,...,Y-B	C45621F,G,...,Z-B

22 tests (C24113D,E,...,Y-B) were not processed because source lines were too long.

17 tests were inapplicable because the implementation does not support SHORT_INTEGER, LONG_INTEGER, other INTEGER types, SHORT_FLOAT, or LONG_FLOAT:

SHORT_INTEGER	C34001D-B, B52004E-AB, B55B09D-AB, C55B07B-AB, B86001CR-AB
LONG_INTEGER	C34001E-B, B52004D-AB, B55B09C-AB, C55B07A-AB, B86001CS-AB
other INTEGER type	B86001DT-AB
SHORT_FLOAT	C34001F-B, C35702A-AB, B86001CP-AB
LONG_FLOAT	C34001G-B, C35702B-AB, B86001CQ-AB

C86001E-B is inapplicable because package SYSTEM is used by package TEXT_IO.

LA3004A0,1,...,6M-AB and LA3004B0,1,...,6M-B are inapplicable because pragma INLINE is not supported.

CE2102D-B, CE2102E-B, CE2102F-B, and CE2102G-B are inapplicable because the implementation does support modes IN_FILE, OUT_FILE, and INOUT_FILE, and also the procedures RESET and DELETE.

CE2106A, CE2107B, CE2107C, CE2107D, CE2107E, CE2108I, CE2108C, CE2110B, CE2111D, CE3111B, CE3111C, CE3114B, and CE3115A, CE3111D, CE3111E, CE3112A, CE3114B, CE3115A are inapplicable because of implementation-dependent characteristics in input-output.

2.4 Implementation Characteristics

One of the purposes of validation is to determine the behavior of a compiler in those areas of the Ada Standard that permit implementations to differ. Class D and E tests specifically check for such implementation differences. However, inapplicable tests in other classes also characterize an implementation. This compiler is characterized by the following interpretations of the Ada Standard:

- Non-graphic characters.

  Non-graphic characters are defined in the ASCII character set but are not permitted in Ada programs, even within character strings. The compiler correctly recognizes these characters as illegal in Ada compilations. The characters in the output listing but are not visible if printed.

- Capacities.

  The compiler correctly processes compilations containing loop statements nested to 65 levels, block statements nested to 65 levels, procedures nested to 17 levels, and 723 variables.

- Universal integer calculations.

  An implementation is allowed to reject universal integer calculations having values that exceed SYSTEM.MAX_INT. This implementation does not reject such calculations and raises NUMERIC_ERROR.

- Universal real calculations.

  An implementation is allowed to reject universal real calculations having values that exceed certain precisions. This implementation does not reject such calculations and processes them correctly.

- Predefined types.

  This implementation does support the predefined types INTEGER, FLOAT, and DURATION. It does not support any other predefined numeric types.

- Based literals.

  An implementation is allowed to reject a based literal with value exceeding SYSTEM.MAX_INT during compilation or it may raise NUMERIC_ERROR during execution. This compiler raises NUMERIC_ERROR during execution.

- Array types.

  An implementation is allowed to raise NUMERIC_ERROR for an array having a 'LENGTH that exceeds STANDARD.INTEGER'LAST and/or SYSTEM.MAX_INT. When an array type is declared with an index range exceeding INTEGER values and with a component that is a null BOOLEAN array, this compiler raises NUMERIC_ERROR when the type is declared.

  When an array type is declared with an index range exceeding SYSTEM.MAX_INT values and with a component that is a null BOOLEAN array, this compiler raises NUMERIC_ERROR when on object of this type is declared.

  A packed BOOLEAN array of length INTEGER_LAST+3 raises NUMERIC_ERROR when the array objects are declared. A packed two-dimensional BOOLEAN array with INTEGER_LAST+3 components raises NUMERIC_ERROR when the array objects are declared.

  A null array with one dimension of length exceeding INTEGER'LAST does not raise NUMERIC_ERROR when the array type is declared or when array objects are assigned.

  In assigning one-dimensional or two-dimensional array types, the entire expression is evaluated before CONSTRAINT_ERROR is raised when checking whether the expression's subtype is comptaible with the target's subtype.

- Discriminated types.

  In assigning record types with discriminants, the entire expression is evaluated before CONSTRAINT_ERROR is raised when checking whether the expression's subtype is compatible with the target's subtype.

  An incompletely declared type with discriminants may be used in an access type definition and constrained either there or in later subtype indications.

- Aggregates.

  When evaluating the choices of a multi-dimensional aggregate all choices are evaluated before checking against the index type.

  When evaluating an aggregate containing subaggregates, all choices are evaluated before being checked for identical bounds.

- Representation clauses.

  'SMALL length clauses are supported.

  Enumeration representation clauses are supported.

- Package CALENDAR.

  TIME_OF and SPLIT are inverses when SECONDS is a non-model number.

- Pragmas.

  Pragma INLINE is not supported for procedures. It is not supported for functions.

- Input/output.

  Package SEQUENTIAL_IO can be instantiated with unconstrained array types and record types with discriminants. Package DIRECT_IO can be instantiated with unconstrained array types and record types with discriminants without defaults. A form parameter is needed in the case of DIRECT_IO and unconstrained array types.

  For SEQUENTIAL_IO, DIRECT_IO and TEXT_IO more than one internal file can be associated with each external file for reading only. An external file associated with more than one internal file cannot be deleted.

  An existing text file can be opened in OUT_FILE mode, can be created in OUT_FILE mode, and can be created in IN_FILE mode.

  Dynamic creation and resetting of a sequential file is allowed.

  Temporary sequential, direct, and text files are not given a name.

- Library tasks.

  Execution of library tasks is discontinued after termination of the main program. This behavior is in accordance with the current LMC interpretation of the LRM (cf. November 1985 LMC meeting).

CHAPTER 3

Compiler Anomalies and Nonconformances

3.1 Anomalies

An anomaly is a test result that, given the pre-validation analysis, was not expected during formal validation but which is judged allowable by the AVF and the AVO under the circumstances of the validation. No anomalies were detected in this validation attempt.

3.2 Nonconformances

Any discrepancy between expected test results and actual test results is considered to be a nonconformance. No nonconformances were detected in this validation attempt.

CHAPTER 4

ADDITIONAL TESTING INFORMATION

4.1 Pre-Validation

Prior to validation, a set of test results for ACVC 1.6 produced by Systeam/German MoD VAX-11 Compiler was submitted to IABG m.b.H. by the applicant for pre-validation review. Analysis of these results demonstrated that the compiler successfully passed all applicable tests.

4.2 Test Site

Tests were compiled and executed at Karlsruhe on two identical host and target configurations.

4.3 Test Tape Information

A test tape containing ACVC Version 1.6 was taken on-site by the validation team. This tape contained all tests applicable to this validation as well as all tests inapplicable to this validation. Tests that make use of values that are specific to the implementation were customized. The test suite was read from tape. The files were structured into directories according to LRM chapters and test categories. The command files were added to the directory, then the files were copied into the second (identical) computer. The whole process took about 3 hours.

4.4 Testing Logistics

Once all tests had been loaded to disk, processing was begun using command scripts provided by Systeam KG Dr. Winterstein. A sample of the text of these scripts is given in Appendix C.

The compiler supports various options that control its operation. The compiler was tested with the following option setting:

List => on,

which causes the compiler to print a listing.

Test were run on two machines simulataneonly starting with the B-Tests. On each machine two parallel batch queues were used. For each chapter two batch jobs were initiated, one for B-Tests, the other for executable tests. Each job initialized its own project library and compiled the report package as far as needed. The report package was modified in

order to print the date and time of test execution. For each test the compiler listing and the result, if any, were written to individual files. These files were written to tape in VAX backup format and archived.

4.5 Testing Duration

The ACVC has not been designed for use in measuring compiler performance. The information reported here thus merely describes the duration of the on-site conformity testing, and is not necessarily an indication of the subject system's performance.

Testing started at 17:24 on 85-11-22 and was completed at 18:09 on 85-11-24. Testing was done on two identical machines. The machines idled on 85-11-24 from about 9 a.m. until 5 p.m. A total of about 72 hours CPU time was consumed.

CHAPTER 5

SUMMARY AND CONCLUSIONS

IABG m.b.H. identified 1873 of the 2162 tests in ACVC version 1.6 as potentially applicable to the Systeam/German MoD VAX-11 Compiler. Excluded were 201 tests requiring too great a floating-point precision, 22 tests with source lines too long, and the 66 withdrawn tests. 41 tests were determined to be inapplicable after they were processed. The remaining 1832 tests were passed by the compiler.

IABG m.b.H. concludes that these results demonstrate acceptable conformance to the Ada Standard.

APPENDIX A

COMPLIANCE STATEMENT

The only allowed implementation dependencies correspond to implementation-dependent pragmas and attributes, to certain machine-dependent conventions as mentioned in Chapter 13 of MIL-STD-1815A, and to certain allowed restrictions on representation classes. The implementation-dependent characteristics of the Systeam/German MoD VAX-11 Compiler are described in the following sections which discuss topics one through eight as stated in Appendix F of the Ada Language Reference Manual (ANSI/MIL-STD-1815A).

What follows is chapter 9 of the Systeam/German MoD VAX-11-Compiler User Manual, where all implementation dependent characteristics of the compiler are described.

9 IMPLEMENTATION-DEPENDENT CHARACTERISTICS

This chapter corresponds to Appendix F of the Ada Language Reference Manual, which describes all implementation-dependent characteristics.

9.1 Implementation-Dependent Pragmas

INTERFACE -
    is implemented for ASSEMBLER

SUPPRESS_ALL -
    causes that all checks that may raise CONSTRAINT_ERROR at run-time are suppressed; this pragma is only allowed at the start of a compilation before the first compilation unit; it applies to the whole compilation

9.2 Implementation-Dependent Attributes

HEAP_ADDRESS -
    applied to an access type yields a value of type ADDRESS (from package SYSTEM). This attribute is only for internal use within the package COLLECTION_MANAGER.

9.3 Specification of the Package SYSTEM

```
PACKAGE system IS

   TYPE address IS PRIVATE;

   TYPE name IS (vax_730, vax_750, vax_780, vax_782);

   system_name   : CONSTANT name := vax_750;
   storage_unit  : CONSTANT := 8;
   memory_size   : CONSTANT := 2 ** 31;
   min_int       : CONSTANT := - 2_147_483_648;
   max_int       : CONSTANT := 2_147_483_647;
   max_digits    : CONSTANT := 9;
   max_mantissa  : CONSTANT := 31;
   fine_delta    : CONSTANT := 2#1.0#E-30;
   tick          : CONSTANT := 0.2E-6;

   SUBTYPE priority IS integer RANGE 0 .. 255;
```

```
   SUBTYPE external_address IS string;

   SUBTYPE byte IS integer RANGE 0..255;

   TYPE    long_word IS ARRAY (0..3) OF byte;
   PRAGMA  PACK (long_word);

   FUNCTION convert_address (addr   : external_address)
                            RETURN address;

   FUNCTION convert_address (addr   : address)
                            RETURN external_address;

   FUNCTION convert_address (addr   : long_word)
                            RETURN address;

   FUNCTION convert_address (addr   : address)
                            RETURN long_word;

   FUNCTION "+"             (addr   : address;
                             offset : integer)
                            RETURN address;

PRIVATE

   -- private declarations

END system;
```

External addresses are represented as strings consisting of hexadecimal digits.

Since the type ADDRESS is private, no representation specifications for objects of this type can be given. If representation specifications for addresses are required, objects of type LONG_WORD can be used to hold address values.

Overloaded functions CONVERT_ADDRESS are defined to allow conversion between the different represenations of addresses.

9.4 Restrictions on Representation Clauses

Address clauses are only implemented for objects.

9.5 Conventions for Implementation-Generated Names

There are no implementation-generated names denoting implementation-dependent components.

9.6 Interpretation of Address Clauses

An object for which an address specification is given must not require an initialization (neiter explicit nor implicit). Otherwise the program is erroneous.

The object starts at the given address. For objects accessed by a descriptor, the descriptor starts at the given address.

9.7 Restrictions on Unchecked Conversions

If

TARGET'SIZE > SOURCE'SIZE

the result value of the unchecked conversion is unpredictable.

9.8 Characteristics of the Input-Output Packages

9.8.1 The NAME Parameter

The string must be a VMS file specification string. The function NAME will return a file specification string (including version number) which is the resultant filename of the file opened or created.

The exception NAME_ERROR is raised if the name parameter is no legal VMS file specification string; for example if it contains illegal characters, is too long or is syntactically incorrect. The file specification string must not contain wild cards even if an unique file is specified; otherwise the exception NAME_ERROR is raised.

In an OPEN operation the execption NAME_ERROR is also raised if the specified file does not exist; in a CREATE operation this exception is raised if the NAME string contains an explicit version number and the specified file does already exist.

9.8.2 The FORM Parameter

9.8.2.1 The Syntax of the FORM string

```
form_parameter ::=
    [ form_specification { , form_specification } ]

form_specification ::= keyword => value

keyword ::= identifier

value   ::= identifier | string_literal | numeric_literal
```

For identifier, numeric_literal, string_literal see LRM Appendix E. Only an integer literal is allowed as numeric_literal (see LRM 2.4).

The exception USE_ERROR is raised if a given FORM paramter string has not the correct syntax or if a condition on a single form specification described in the following sections is not fulfilled.

9.8.2.2 General Form Specifications

In the following the form specifications which are allowed for all files are described.

- ALLOCATION => numeric_literal

This value specifies the number of blocks which are allocated initially, it is only used in a create operation and ignored in an open operation. The value of allocation in the form string returned by the function form specifies the initial allocation size for existing files too.

- EXTENSION => numeric_literal

This value specifies the number of blocks by which a file is extended if necessary; the value 0 means that the RMS default value is taken. For existing files this value is only used for processing between an open and a close operation.

For details see the VAX-11 / RMS Reference Manual.

9.8.3 Text_IO

9.8.3.1 Implementation_Dependent_Types

The implementation depencent types COUNT and FIELD defined in the package specification of TEXT_IO have the following upper bounds :

COUNT'LAST = 2_147_483_647 (= INTEGER'LAST)

FIELD'LAST = 255

9.8.3.2 Text_Files

Text files are represented as sequential files with variable record format. One line is represented as a sequence of one or more records; all records except from the last one have a continuation marker (ASCII.LF) as last character which does not belong to the line.

A line terminator which is not followed by a page terminator is not represented explicitly in the external file. A line terminator followed by a page terminator is represented as an ASCII.FF behind the last character of the last line of a page, i.e. the last character of the last record which belongs to the last line of the page is ASCII.FF. A line terminator followed by a page terminator followed by a file terminator is not represented explicitly in the external file; the combination of these three terminators is indicated by the end of the file. For input from terminal the combination of a line terminator followed by a page terminator followed by a file terminator is represented as ASCII.SUB (= CTRL Z) behind the last character of the file.

In the following the form specifications which are only allowed for text files are described.

Only for output files :

- MAX_RECORD_SIZE => numeric_literal

This value specifies the maximum length of a record in the external file. Each record with a continuation marker has exactly this maximum record length. The value must be in the range from 2 up to 512. If the value is specified for an existing file it must confirm with the value of the external file.

- END_OF_FILE

If the keywork END_OF_FILE is specified for an existing file in an open for an output file then the file is opened at the end of the file; i.e. the existing file is extended and not rewritten. This keyword is only allowed for an output file; it has only an effect in an open operation and is ignored in a create.

Only for input files :

- PROMPTING => string_literal

This string is output on the terminal before an input record is read if the input file is associated with a terminal, otherwise this form specification is ignored.

The default form string for an input text file is :

"ALLOCATION => 3, EXTENSION => 0, PROMPTING => """" "

The default form string for an output text file is :

"ALLOCATION => 3, EXTENSION => 0, MAX_RECORD_SIZE => 512"

- CHARACTER_IO

In addition to the input/output facilities with record structurec external files another form of input/output is provided for text files. It is possible to transfer single characters from/to a terminal device. This form of input/output is specified by the keyword CHARACTER_IO in the form string. If character i/o is specified, no other form specification is allowed and the file name must denote a terminal device.

For an infile the external file (associated with a terminal) is regarded to contain a single line. An ASCII.SUB (= CTRL Z) character represents ar line terminator followed by a page terminator followed by a file terminator. Arbitrary characters (including all control characters except from ASCII.SUB) may be read; a character read is not echoed to the terminal.

For an outfile arbitrary characters (including all control characters and escape sequences) may be written on the external

file (terminal). A line terminator is represented as ASCII.CR followed by ASCII.LF, a page terminator is represented as ASCII.FF and a file terminator is not represented on the external file.

9.8.3.3 Standard_Files

The standard input (resp. output) file is associated with SYS$INPUT (resp. SYS$OUTPUT). If a program reads from the standard input file, the logical name SYS$INPUT must denote an existing file. If a program writes to the standard output file, a file with the logical name SYS$OUTPUT is created, if no such file exists, otherwise the existing file is extended.

The qualifiers /INPUT and /OUTPUT may be used for the VMS RUN command to associate VMS files with the standard files of TEXT_IO.

The name and form strings for the standard files are :

```
standard_input  : NAME => "SYS$INPUT:"
                  FORM => "PRCMPTING => ""*""    "

standard_output : NAME => "SYS$OUTPUT:"
                  FORM => "MAX_RECORD_SIZE => 512"
```

9.8.4 Sequential_and_Direct_Files

Sequential and direct files are represented by RMS sequential, relative or indexed files with fixed-length or variable-length records. Each element of the file is stored in one record.

9.8.4.1 Restrictions_Concerning_the_ELEMENT_TYPE

- input/output of access types is not defined.
- the attribute ADDRESS applied to an object of the element type must specify the start address of the value of the object (not the address of a descriptor).
- input/output is not possible for an object whose start address is not byte aligned (may only occur if a representation specification is given).

- the attribute SIZE applied to an object of the element type must deliver the number of bits allocated contiguously in the memory for the object; this value must be a multiple of SYSTEM.STORAGE_UNIT. For example objects of record types with dynamic array components are not stored contiguously.

- if a fixed record format is used all objects to be input or output must have the same size (ELEMENT_TYPE'SIZE).

- input/output of elements of an unconstrained array type is only possible for files with variable-length records.

- for RMS sequential [relative] files the size of an object to be input or output must not be greater than 32767 [16383].

9.8.4.2 Sequential_Files

A sequential file is represented by a RMS sequential file with either fixed-length or else variable-length records which may be specified by the form parameter.

- MAX_RECORD_SIZE => numeric_literal

This value specifies the maximum record size in bytes; the value 0 indicates that there is no limit. This form specification is only allowed for files with variable record format. If the value is specified for an existing file it must confirm with the value of the external file. For files with fixed-length records the maximum record size equals ELEMENT_TYPE'SIZE / SYSTEM.STORAGE_UNIT.

- RECORD_FORMAT => VARIABLE | FIXED

by this form specification the record format may be specified. If the format is specified for an existing file it must equal the format of the external file.

- END_OF_FILE

If the keywork END_OF_FILE is specified for an existing file in an open for an output file then the file is opened at the end

of the file; i.e. the existing file is extended and not rewritten. This keyword is only allowed for an output file; it has only an effect in an open operation and is ignored in a create.

The default form string for a sequential file is :

```
"ALLOCATION     => 3,          EXTENSION        => 0, " &
"RECORD_FORMAT => VARIABLE, MAX_RECORD_SIZE => 0  "
```

9.8.4.3 Direct_Files

The implementation dependent type COUNT defined in the package specification of DIRECT_IO has an upper bound of :

```
COUNT'LAST = 2_147_483_647 (= INTEGER'LAST)
```

Direct files are represented by RMS sequential files with fixed-length records or by relative or indexed files with either fixed-length or else variable-length records. For indexed files the record index is stored as unsigned four bytes binary value in the first four bytes of each record. If not explicitly specified otherwise the maximum record size equals ELEMENT_TYPE'SIZE / SYSTEM.STORAGE_UNIT.

- BUCKET_SIZE => numeric_literal

This value specifies the number of blocks (one block is 512 bytes) for one bucket; the value 0 means that the value is evaluated by RMS to the minimal number of blocks which is necessary to contain one record. The value must be in the range from 0 up to 32. This form specification is only allowed for relative or indexed files. If the value is specified for an existing file it must confirm with the value of the external file.

- MAX_RECORD_SIZE => numeric_literal

This value specifies the maximum record size in bytes. The value 0 which indicates that there is no limit is only allowed for indexed files. A positive value must be greater or equal to ELEMENT_TYPE'SIZE / SYSTEM.STORAGE_UNIT. This form specification is only allowed for files with variable record

format. If the value is specified for an existing file it must confirm with the value of the external file.

- RECORD_FORMAT => VARIABLE | FIXED

By this form specification the record format may be specified. If the format is specified for an existing file it must equal the format of the external file.

- ORGANIZATION => INDEXED | RELATIVE | SEQUENTIAL

By this form specification the file organization may be specified. If the organization is specified for an existing file it must equal the organization of the external file.

The default form string for a direct file is :

```
"ALLOCATION   => 3,          EXTENSION      => 0,    " &
"ORGANIZATION => SEQUENTIAL, RECORD_FORMAT => FIXED"
```

9.8.5 General_Limitations

The total number of open files (including the two standard files) must not be greater than 18. An attempt to exceed this limit raises the exception USE_ERROR.

9.8.6 File_Sharing

The only form of file sharing which is allowed is shared reading. If two or more files are associated with the same external file at one time (regardless if these files are declared in the same program or task) all of these (internal) files must be opened with the mode IN_FILE. An attempt to open one of these files with another mode than IN_FILE will raise the exception USE_ERROR.

Files associated with terminal devices (which is only legal for text files) are excepted from this restriction. Such files may be opened with an arbitrary mode at the same time and associated with the same terminal device.

9.3.7 Exceptions_in_Input-Output

Besides the situations described in 9.8.1 and 9.8.2 under which NAME_ERROR and USE_ERROR may be raised in the following additional conditions are listed under which one of the exceptions NAME_ERROR, USE_ERROR, DEVICE_ERROR or DATA_ERROR is raised.

The exception USE_ERROR is raised if the characteristics of the external file are not appropriate for the file type; for example if the record size of a file with fixed-length records does not correspond to the size of the element type of a direct_io or sequential_io file. USE_ERROR is raised also if the function NAME is applied to a temporary file.

In general it is only guaranteed that a file which is created by an Ada program may be reopened by another program, if the file types and the form strings are the same.

The exception DEVICE_ERROR is never raised. Instead of this exception the exception USE_ERROR is raised whenever an error occured during an operation of the underlying RMS system. This may happen if an internal error was detected, an operation is not possible for reasons depending on the file or device characteristics, a size restriction is violated, a capacity is exceeded or for similar reasons.

The exception DATA_ERROR is raised by the procedure READ if the size of the element in the external file to be read differs from the storage size of the given variable; this may only happen if a variable record size is used. This exception is raised too if an element with the specified position in a direct file does not exist; this is only possible if the file is associated with a relative or an indexed file.

In general the exception DATA_ERROR is not raised by the procedure READ if the element read is not a legal value of the element type.

9.8.8 Specification of the Package LOW_LEVEL_IO

```
PACKAGE low_level_io IS

   TYPE device_type IS (null_device);

   TYPE data_type IS
      RECORD
         NULL;
      END RECORD;

   PROCEDURE send_control    (device : device_type;
                              data   : IN OUT data_type);

   PROCEDURE receive_control (device : device_type;
                              data   : IN OUT data_type);

END low_level_io;
```

9.9 Requirements for Main Programs

The main program must be a parameterless library procedure.

9.10 Specification of the Package COLLECTION_MANAGER

```
GENERIC
   TYPE elem IS PRIVATE;
   TYPE acc  IS ACCESS elem;
   size : integer := 100;
PACKAGE collection_manager IS

   PROCEDURE mark;

      -- Mark the heap of type ACC

   PROCEDURE release;

      -- Deallocate all objects on the heap of ACC which were
      -- allocated after the last MARK operation for that heap.
      -- RELEASE without previous MARK raises CONSTRAINT_ERROR

   PROCEDURE reset;

      -- Deallocate all objects on the heap of ACC

END collection_manager;
```

The difference between the number of calls of the procedures MARK and RELEASE must be in the range 0 .. SIZE. After a call of RESET the effect of all previous calls of MARK and RELEASE is canceled. The counting of the difference mentioned above starts from 0.

The value delivered by the attribute STORAGE_SIZE applied to the actual type for ACC is meaningless if the Collection Manager is used.

9.11 Other_Characteristics

9.11.1 Source_Programs

The maximum line length is 80. Longer lines are cut and an error is reported.

9.11.2 Program_Library

The maximum number of units contained in a program library is 2_000. The maximum number of imported units for one compilation unit is 63.

9.11.3 ADDRESS_and_PRIORITY

The package SYSTEM must be named by a with clause of a compilation unit, if the predefined attribute ADDRESS or the predefined pragma PRIORITY is used within that unit.

9.11.4 Storage_for_Tasks

The memory space reserved for a task is 4K byte. If a task has inner tasks, a length clause must be given at least for the enclosing task. At least 4K byte should be provided for each inner task. The activation of a small task requires about 1.1K byte.

APPENDIX B

TEST PARAMETERS

Certain tests in the ACVC make use of implementation-dependent values, such as the maximum length of an input line and invalid file names. A test that makes use of such values is identified by the extension .TST in its file name. Actual values to be substituted are identified by names that begin with a dollar sign. A value is substituted for each of these names before the test is run. The values used for this validation are given below.

```
-- MACRO.DEFS

-- THIS FILE CONTAINS THE MACRO DEFINITIONS USED IN THE ACVC TESTS.
-- THESE DEFINITIONS ARE FOR THE PRELIMINARY ADA TEST TRANSLATOR.


-- THE MAXIMUM INPUT LINE LENGTH PERMITTED BY THE IMPLEMENTATION.
-- USED IN: COMMENTS IN THIS FILE
MAX_IN_LEN      80

-- TWO IDENTIFIERS THAT ARE MAX_IN_LEN LONG AND DIFFER ONLY IN THEIR
-- LAST CHARACTER.
-- USED IN: C23003A, B23003D, B23003E, B23003F
BIG_ID1 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA1
BIG_ID2 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA2

-- TWO IDENTIFIERS THAT ARE MAX_IN_LEN LONG AND DIFFER ONLY IN THEIR
-- MIDDLE CHARACTER.
-- USED IN: C23003A
BIG_ID3 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA1AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BIG_ID4 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA2AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

-- A BASED_INTEGER LITERAL (PREFERABLY BASE 8 OR 16) WHOSE HIGHEST ORDER
-- NON-ZERO BIT WOULD FALL IN THE SIGN BIT POSITION OF THE
-- REPRESENTATION FOR SYSTEM.MAX_INT, I.E., AN ATTEMPT TO WRITE A
-- NEGATIVE VALUED LITERAL SUCH AS -2.
-- USED IN: E24101A
NEG_BASED_INT   16#FFFFFFFE#

-- AN INTEGER LITERAL OF VALUE 298 WITH ENOUGH LEADING ZEROES SO THAT IT
-- IS MAX_IN_LEN CHARACTERS LONG.
-- USED IN: C24003A
BIG_INT_LIT       00000000000000000000000000000000000000000000000000000000000000000000000000000298

-- A REAL LITERAL THAT CAN BE EITHER OF FLOATING OR FIXED POINT TYPE,
-- HAS VALUE 690.0, AND HAS ENOUGH LEADING ZEROES TO BE MAX_IN_LEN
-- CHARACTERS LONG.
-- USED IN: C24003B, C24003C
BIG_REAL_LIT      0000000000000000000000000000000000000000000000000000000000000000000000000069.0E1

-- A STRING LITERAL CONTAINING ALL THE ASCII CHARACTERS WITH PRINTABLE
-- GRAPHICS THAT ARE NOT IN THE BASIC 55 ADA CHARACTER SET.
-- USED IN: A26004A
EXTENDED_ASCII_CHARS     "abcdefghijklmnopqrstuvwxyz!$%?@[\]^`{}~"

-- AN ENUMERATED_TYPE_DEFINITION FOR A CHARACTER TYPE WHOSE LITERALS ARE
-- THE IDENTIFIER NON_NULL AND ALL NON-ASCII CHARACTERS WITH PRINTABLE
-- GRAPHICS, E.G., CENT-SIGN.
-- USED IN: A26004A
NON_ASCII_CHAR_TYPE      (NON_NULL)

-- A SEQUENCE OF BLANKS THAT IS MAX_IN_LEN - 20 LONG.
-- USED IN: B22001A, B22001B, B22001C, B22001D, B22001E, B22001F,
--            B22001G, B22001I, B22001J, B22001K, B22001L, B22001M,
--            B22001N
BLANKS

-- THE MAXIMUM DIGITS SUPPORTED FOR FLOATING POINT TYPES.
-- USED IN: B35701A
MAX_DIGITS      9

-- A NAME OF A PREDEFINED NUMERIC TYPE OTHER THAN FLOAT, INTEGER,
-- SHORT_FLOAT, SHORT_INTEGER, LONG_FLOAT, OR LONG_INTEGER.
-- (MOST IMPLEMENTATIONS WILL NOT HAVE ANY SUCH TYPES.)
-- USED IN: B86001DT
NAME    $NAME

-- THE UNIVERSAL_INTEGER LITERAL_EXPRESSIONS WHOSE VALUES ARE
```

```
-- INTEGER'FIRST AND INTEGER'LAST.
-- USED IN: B548018
INTEGER_FIRST    -2147483648
INTEGER_LAST     2147483647

-- A UNIVERSAL_REAL VALUE (NOT SUBJECT TO ROUND-OFF ERROR IF POSSIBLE)
-- THAT LIES BETWEEN DURATION'BASE'FIRST AND DURATION'FIRST.  IF NO SUCH
-- VALUES EXIST, ANY VALUE IN THE RANGE OF DURATION WILL DO.
-- USED IN: C960058
LESS_THAN_DURATION       -0.0

-- A UNIVERSAL_REAL VALUE (NOT SUBJECT TO ROUND-OFF ERROR IF POSSIBLE)
-- THAT LIES BETWEEN DURATION'BASE'LAST AND DURATION'LAST.  IF NO SUCH
-- VALUES EXIST, ANY VALUE IN THE RANGE OF DURATION WILL DO.
-- USED IN: C960058
GREATER_THAN_DURATION    0.0

-- UNIVERSAL_REAL VALUES THAT ARE LESS THAN DURATION'BASE'FIRST AND
-- GREATER THAN DURATION'BASE'LAST, RESPECTIVELY.
-- USED IN: C96005C
LESS_THAN_DURATION_BASE_FIRST    -200_000.0
GREATER_THAN_DURATION_BASE_LAST  200_000.0

-- THE VALUE OF COUNT'LAST IN TEXT_IO PACKAGE.
-- USED IN: CE3002B
COUNT_LAST       2147483647

-- THE VALUE OF FIELD'LAST IN TEXT_IO PACKAGE.
-- USED IN: CE3002C
FIELD_LAST       255

-- AN ILLEGAL EXTERNAL FILE NAME THAT EITHER (PREFERABLY) CONTAINS
-- INVALID CHARACTERS OR IS TOO LONG.
-- USED IN: CE2102C
FILE_NAME_WITH_BAD_CHARS         abc!def.dat

-- AN EXTERNAL FILE NAME THAT EITHER (PREFERABLY) CONTAINS A WILD CARD
-- CHARACTER OR IS TOO LONG.
-- USED IN: CE2102C
FILE_NAME_WITH_WILD_CARD_CHAR    abc*def.dat

-- AN ILLEGAL EXTERNAL FILE NAME (E.G., TOO LONG, OR CONTAINING INVALID
-- CHARACTERS.
-- USED IN: CE2103A, CE2103B, CE3102B, CE3107A
ILLEGAL_EXTERNAL_FILE_NAME1      xsfyz.dat
ILLEGAL_EXTERNAL_FILE_NAME2      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

APPENDIX C

COMMAND SCRIPTS

What follows is

- the command script, which compiles and executes the report package and the report test routines.

- the sequence of commands which implements the linkandgo-procedure.

The other command procedures work in exactly the same manner.

```
$! p1 : compiler version
$! p2 : additional parameter for compiler
$ version = p1
$ if p2 .eqs. "" then p2="options=list=>on"
$ env = f$environment("PROCEDURE")
$ acvcversion = f$parse(env,,,"DEVICE") + f$parse(env,,,"DIRECTORY")
$ acvc  = acvcversion - "]" + ".-]"
$ acvcsupport = acvcversion - "]" + ".support]"
$!
$ @'version'createlib
$ @'version'compile 'acvcsupport'repspec
$ @'version'compile 'acvcsupport'reptocy
$ @'version'compile 'acvcsupport'checkfile
$ @'version'compile 'acvcsupport'varstrspc
$ @'version'compile 'acvcsupport'varstrbod
$!
$ @'version'compile 'acvcsupport'CZ1101A.ADA  'p2'
$ @'acvc'linkandgo CZ1101A 'version'
$ @'version'compile 'acvcsupport'CZ1102A.ADA  'p2'
$ @'acvc'linkandgo CZ1102A 'version'
$ @'version'compile 'acvcsupport'CZ1103A.ADA  'p2'
$ @'acvc'linkandgo CZ1103A 'version'
$ @'version'compile 'acvcsupport'CZ1201A.ADA  'p2'
$ @'acvc'linkandgo CZ1201A 'version'
$ @'version'compile 'acvcsupport'CZ1201B.ADA  'p2'
$ @'acvc'linkandgo CZ1201B 'version'
$ @'version'compile 'acvcsupport'CZ1201C.ADA  'p2'
$ @'acvc'linkandgo CZ1201C 'version'
$ @'version'compile 'acvcsupport'CZ1201D.ADA  'p2'
$ @'acvc'linkandgo CZ1201D 'version'
$ @'version'deletelib
```

```
$ veri = f$verify(0)
$!*************************************************************************
$! Command procedure to link and execute one module of
$! the acvc's.
$! p1 = name of test
$! p2 = compiler version
$!*************************************************************************
$!
$ testname    = p1
$ version     = p2
$!
$ on error then goto error_exit
$ @'version'link 'testname' 'testname'
$ create 'testname'.res
$ define/user sys$output 'testname'.res
$ run 'testname'
$ delete 'testname'.exe;
$ veri = f$verify(veri)
$ exit
$ error_exit :
$ write sys$output ">>> 'linkandgo' not terminated normally "
$ veri = f$verify(veri)
$ exit
```

APPENDIX D

COMPLETE LIST OF TESTS AND RESULTS

This Appendix presents a complete list of the ACVC test files used in the validation attempt, presented in order by ACVC Implementers' Guide section and objective. Each test name indicates the class of the test and which test objective in the ACVC Implementers' Guide applies to the test.

Each test has a name that identifies the section of the Ada Standard addressed by the test objective. The name of a test is interpreted according to the table below, where the first column indicates the character position in the name and the second column, the meaning of that position:

POS	MEANING
1	Test class: A, B, C, D, E, L.
2	Implementers' Guide chapter number (in hexadecimal).
3	Implementers' Guide section number within a chapter (in hexadecimal)
4	Implementers' Guide subsection number (in hexadecimal)
5-6	Implementers' Guide Test Objective number (in decimal)
7	Test sequence letter
8	Optional Compilation sequence digit or letter
9	Optional Main program designator in the case of a test having multiple compilation units.

Characters 8 and 9 are only present for tests that consist of several separately compiled units. A series of separately compiled units is counted as one test for reporting purposes. The eighth character indicates the order in which the units are to be compiled, with unit 0 being compiled first. The ninth character is only present for a file containing a main program for a test comprising multiple files and is always M.

The suffix -AB means the test was written prior to release of the ANSI Standard and is also valid for the version of Ada published in July 1980. The suffix -B means the test was written specifically for the ANSI Standard. Tests without a suffix have not yet had their names revised to -AB.

A file name ending with the extension .TST indicates that the test depends on one or more of the implementation-

dependent parameters listed in Appendix B. A file name ending with .DEP indicates that the test is not necessarily applicable to all implementations because it depends upon the support of language features that a compiler may legally not implement.

The result for each file in ACVC Version 1.6 is given in the following pages, where:

P indicates Passed.

F indicates Failed.

N/A indicates Not Applicable to this implementation.

W indicates Withdrawn due to test errors.

A test may comprise several separate compilation units contained in two or more files; the names of such files are indented under the name of the test. The letter 'M' indicates which of these files contains the main procedure.

Support Units

CHECK_FILE-B.ADA	P
REPORT_SPEC-AB.ADA	P
REPORT_BODY-B.ADA	P
VAR_STRINGS_SPEC.ADA	P
VAR_STRINGS_BODY.ADA	P
CZ1101A-AB.ADA	P
CZ1102A-AB.ADA	P
CZ1103A-B.ADA	P
CZ1201A-AB.ADA	P
CZ1201B-AB.ADA	P
CZ1201C-AB.ADA	P
CZ1201D-AB.ADA	P

ACVC 1.6 test results chapter 2 IAbG - AVF

A21003A.ADA	P	C24113G-B.DEP	N/A
B22001A-AB.TST	P	C24113H-B.DEP	N/A
B22001B-AB.TST	P	C24113I-B.DEP	N/A
B22001C-AB.TST	P	C24113J-B.DEP	N/A
B22001D-AB.TST	P	C24113K-B.DEP	N/A
B22001E-AB.TST	P	C24113L-B.DEP	N/A
B22001F-AB.TST	P	C24113M-B.DEP	N/A
B22001G-AB.TST	P	C24113N-B.DEP	N/A
B22001H-AB.ADA	P	C24113O-B.DEP	N/A
B22001I-AB.TST	P	C24113P-B.DEP	N/A
B22001J-AB.TST	P	C24113Q-B.DEP	N/A
B22001K-AB.TST	P	C24113R-B.DEP	N/A
B22001L-AB.TST	P	C24113S-B.DEP	N/A
B22001M-AB.TST	P	C24113T-B.DEP	N/A
B22001N-AB.TST	P	C24113U-B.DEP	N/A
A22002A.ADA	P	C24113V-B.DEP	N/A
B22003A.ADA	P	C24113W-B.DEP	N/A
B22004A.ADA	P	C24113X-B.DEP	N/A
B22004B.ADA	P	C24113Y-B.DEP	N/A
B22004C.ADA	P	B26002A.ADA	P
C23001A.ADA	P	C26002B.ADA	P
B23002A.ADA	P	A26004A.TST	P
C23003A.TST	P	B26005A.ADA	P
B23003D-AB.TST	P	C26006A-AB.ADA	P
B23003E-AB.TST	P	C26006A-AB.ADA	P
B23003F-AB.TST	P	C27001A-AB.ADA	P
B23004A.ADA	P	C27002A-B.ADA	P
B23004B.ADA	P	B29001A-B.ADA	P
B24001A.ADA	P	A29002A-B.ADA	P
B24001B.ADA	P	A29002B-B.ADA	P
B24001C.ADA	P	A29002C-B.ADA	P
C24002A.ADA	P	A29002D-B.ADA	P
C24002B.ADA	P	A29002E-B.ADA	P
C24002C.ADA	P	A29002F-B.ADA	P
C24003A.TST	P	A29002G-B.ADA	P
C24003B.TST	P	A29002H-B.ADA	P
C24003C.TST	P	A29002I-B.ADA	P
B24005A.ADA	P	A29002J-B.ADA	P
B24005B.ADA	P	C29002K-B.ADA	P
B24101A-B.TST	P		
C24102A.ADA	P		
C24102B.ADA	P		
C24102C.ADA	P		
C24103A.ADA	P		
B24104A.ADA	P		
B24104B.ADA	P		
B24104C.ADA	P		
C24113A-B.DEP	P		
C24113B-B.DEP	P		
C24113C-B.DEP	P		
C24113D-B.DEP	N/A		
C24113E-B.DEP	N/A		
C24113F-B.DEP	N/A		

B32103A-AB.ADA	P	B35701A.TST	P
B32106A-B.ADA	P	C35702A-AB.DEP	N/A
C32107B-B.ADA	P	C35702B-AB.DEP	N/A
B32201A-B.ADA	P	C35703A.ADA	P
B32202A-B.ADA	P	C35704A-AB.ADA	P
B32202B-B.ADA	P	C35704B-AB.ADA	P
B32202C-B.ADA	P	C35704C-AB.ADA	P
C32203A-B.ADA	P	C35704D-AB.ADA	P
A32203B-B.ADA	P	C35705A-B.DEP	P
A32203C-B.ADA	P	C35705B-B.DEP	P
A32203D-B.ADA	P	C35705C-B.DEP	P
B33001A.ADA	P	C35705D-B.DEP	P
B33002A.ADA	P	C35705E-B.DEP	P
B33003A.ADA	P	C35705F-B.DEP	N/A
B33003B-AB.ADA	P	C35705G-B.DEP	N/A
B33003C-AB.ADA	P	C35705H-B.DEP	N/A
B33004A.ADA	P	C35705I-B.DEP	N/A
B33006A-B.ADA	P	C35705J-B.DEP	N/A
C34001A-B.ADA	P	C35705K-B.DEP	N/A
C34001B-B.ADA	P	C35705L-B.DEP	N/A
C34001C-B.ADA	P	C35705M-B.DEP	N/A
C34001D-B.DEP	N/A	C35705N-B.DEP	N/A
C34001E-B.DEP	N/A	C35705O-B.DEP	N/A
C34001F-B.DEP	N/A	C35705P-B.DEP	N/A
C34001G-B.DEP	N/A	C35705Q-B.DEP	N/A
C34001H-B.ADA	P	C35705R-B.DEP	N/A
C34001I-B.ADA	P	C35705S-B.DEP	N/A
C34001K-B.ADA	P	C35705T-B.DEP	N/A
C34001L-B.ADA	P	C35705U-B.DEP	N/A
C34001M-B.ADA	P	C35705V-B.DEP	N/A
C34001N-B.ADA	P	C35705W-B.DEP	N/A
C34001O-B.ADA	P	C35705X-B.DEP	N/A
C34001P-B.ADA	P	C35705Y-B.DEP	N/A
C34001Q-B.ADA	P	C35706A-B.DEP	P
C34001R-B.ADA	P	C35706B-B.DEP	P
B34001S-AB.ADA	P	C35706C-B.DEP	P
C34001T-B.ADA	P	C35706D-B.DEP	P
C34002A-B.ADA	P	C35706E-B.DEP	P
C34002B-B.ADA	P	C35706F-B.DEP	N/A
B34006A-B.ADA	P	C35706G-B.DEP	N/A
A34006B-B.ADA	P	C35706H-B.DEP	N/A
B35101A.ADA	P	C35706I-B.DEP	N/A
C35104A.ADA	P	C35706J-B.DEP	N/A
B35301A.ADA	P	C35706K-B.DEP	N/A
B35501A.ADA	P	C35706L-B.DEP	N/A
C35504A-AB.ADA	P	C35706M-B.DEP	N/A
C35504B-B.ADA	P	C35706N-B.DEP	N/A
C35505A.ADA	P	C35706O-B.DEP	N/A
C35505B.ADA	P	C35706P-B.DEP	N/A
B35506A.ADA	P	C35706Q-B.DEP	N/A
B35506B.ADA	P	C35706R-B.DEP	N/A
C35506A-AB.ADA	P	C35706S-B.DEP	N/A
C35506B-B.ADA	P	C35706T-B.DEP	N/A

C35706U-B.DEP	N/A	C35706X-B.DEP	N/A
C35706V-B.DEP	N/A	C35706Y-B.DEP	N/A
C35706W-B.DEP	N/A	B35709A.ADA	P
C35706X-B.DEP	N/A	C35711A-B.ADA	P
C35706Y-B.DEP	N/A	C35802A-B.DEP	P
C35707A-B.DEP	P	C35802B-B.DEP	P
C35707B-B.DEP	P	C35802C-B.DEP	P
C35707C-B.DEP	P	C35802D-B.DEP	P
C35707D-B.DEP	P	C35802E-B.DEP	P
C35707E-B.DEP	P	C35802F-B.DEP	N/A
C35707F-B.DEP	N/A	C35802G-B.DEP	N/A
C35707G-B.DEP	N/A	C35802H-B.DEP	N/A
C35707H-B.DEP	N/A	C35802I-B.DEP	N/A
C35707I-B.DEP	N/A	C35802J-B.DEP	N/A
C35707J-B.DEP	N/A	C35802K-B.DEP	N/A
C35707K-B.DEP	N/A	C35802L-B.DEP	N/A
C35707L-B.DEP	N/A	C35802M-B.DEP	N/A
C35707M-B.DEP	N/A	C35802N-B.DEP	N/A
C35707N-B.DEP	N/A	C35802O-B.DEP	N/A
C35707O-B.DEP	N/A	C35802P-B.DEP	N/A
C35707P-B.DEP	N/A	C35802Q-B.DEP	N/A
C35707Q-B.DEP	N/A	C35802R-B.DEP	N/A
C35707R-B.DEP	N/A	C35802S-B.DEP	N/A
C35707S-B.DEP	N/A	C35802T-B.DEP	N/A
C35707T-B.DEP	N/A	C35802U-B.DEP	N/A
C35707U-B.DEP	N/A	C35802V-B.DEP	N/A
C35707V-B.DEP	N/A	C35802W-B.DEP	N/A
C35707W-B.DEP	N/A	C35802X-B.DEP	N/A
C35707X-B.DEP	N/A	C35802Y-B.DEP	N/A
C35707Y-B.DEP	N/A	C35904A-B.ADA	W
C35708A-B.DEP	P	B35A03A-B.ADA	P
C35708B-B.DEP	P	B36101A-AB.ADA	P
C35708C-B.DEP	P	B36102A.ADA	P
C35708D-B.DEP	P	B36103A.ADA	P
C35708E-B.DEP	P	B36105A-B.ADA	P
C35708F-B.DEP	N/A	B36171A-B.ADA	P
C35708G-B.DEP	N/A	B36171B-B.ADA	P
C35708H-B.DEP	N/A	B36171C-AB.ADA	P
C35708I-B.DEP	N/A	B36171D-AB.ADA	P
C35708J-B.DEP	N/A	B36171E-AB.ADA	P
C35708K-B.DEP	N/A	B36171F-AB.ADA	P
C35708L-B.DEP	N/A	B36171G-AB.ADA	P
C35708M-B.DEP	N/A	B36171H-AB.ADA	P
C35708N-B.DEP	N/A	B36171I-AB.ADA	P
C35708O-B.DEP	N/A	C36172A-B.ADA	P
C35708P-B.DEP	N/A	C36174A-B.ADA	P
C35708Q-B.DEP	N/A	B36201A-B.ADA	P
C35708R-B.DEP	N/A	B36202A-B.ADA	P
C35708S-B.DEP	N/A	B36202B-B.ADA	P
C35708T-B.DEP	N/A	C36204A-B.ADA	P
C35708U-B.DEP	N/A	C36205A.ADA	P
C35708V-B.DEP	N/A	C36205B.ADA	P
C35708W-B.DEP	N/A	C36205C.ADA	P

C362C5D.ADA P
C362C5E.ADA P
C362C5F.ADA P
C362C5G.ADA P
C362C5H.ADA P
C362C5I.ADA P
C362C5J.ADA P
C362C5K.ADA P
C363C1A-B.ADA P
C363C1B-AB.ADA P
C363C2A.ADA P
C363C3A.ADA P
C363C4A-B.ADA P
C363C5A-AB.ADA P
B37C03A-AB.ADA P
B37C04A-B.ADA P
B37C04B-E.ADA P
B37C04C-B.ADA P
B37C04D-B.ADA P
B37C04E-B.ADA P
B37C04F-B.ADA P
B37C04G-B.ADA P
C37C05A.ADA P
C37C07A-AB.ADA P
C37C08A-B.ADA P
C37C08B-B.ADA P
C37C11A-B.ADA P
C37C12A-AB.ADA P
C37C13A-AB.ADA P
B37101A.ADA P
C37103A-AB.ADA P
C37105A.ADA P
B372C1A.ADA P
B372C2A.ADA P
B372C2B.ADA P
B372C3A.ADA P
B372C4A-AB.ADA P
B372C5A-AB.ADA P
C372C8A-B.ADA P
C372C8B-AB.ADA P
C372C9A.ADA P
B373C1A.ADA P
B373C1B.ADA P
B373C2A-AB.ADA P
B373C3A.ADA P
C373C4A-AB.ADA P
C373C5A.ADA P
C373C8A.ADA P
C373C7A-AB.ADA P
B373C7B-AB.ADA P
C373C9A-AB.ADA P
B373C9B-AB.ADA P
C3731CA-AB.ADA P

B3731CB-E.ADA P
B37311A-AB.ADA P
B38001A.ADA P
B38003A-AB.ADA P
C38004A.ADA P
C38005A-B.ADA P
C38006A-B.ADA P
C38007A-B.ADA P
B38008A-B.ADA P
B38008B-AB.ADA P
B38101A-B.ADA P
B38101B-AB.ADA P
C38102A-AB.ADA P
C38102B-B.ADA P
C38102C-B.ADA P
B38103A-B.ADA P
B3810BB-B.ADA P
B38103C0-B.ADA P
B38103C1-B.ADA P
B38103C2-B.ADA P
B38103C3M-B.ADA P
B3810CA-B.ADA P
B3810CA-AB.ADA P
B3810CB-AB.ADA P
B3810CA-B.ADA P
B3810CB-B.ADA P
A3810CD-B.ADA P
A3810CE-B.ADA P

B41101A-B.ADA	P	C43106B-B.ADA	P
B41101C-AB.ADA	P	C43107A-B.ADA	P
C41101D-B.ADA	P	B43201A-B.ADA	P
B41102A-AB.ADA	P	B43201B-B.ADA	P
B41102B-B.ADA	P	B43201C-B.ADA	P
B41102C-B.ADA	P	B43201D-B.ADA	P
C41103A-B.ADA	P	B43202A-B.ADA	P
C41103B-B.ADA	P	B43202B-B.ADA	P
C41105A-B.ADA	P	B43202C-B.ADA	P
C41106A-B.ADA	P	B43203A-B.ADA	P
C41107A-AB.ADA	P	B43203B-B.ADA	P
B41201A-B.ADA	P	C43205A-B.ADA	P
B41201C.ADA	P	C43205B-B.ADA	P
C41201D-B.ADA	P	C43205C-B.ADA	P
B41202A-B.ADA	P	C43205D-B.ADA	P
B41202B-AB.ADA	P	C43205E-B.ADA	P
B41202C-B.ADA	P	C43205F-B.ADA	P
B41202D-B.ADA	P	C43205G-B.ADA	P
C41203A-B.ADA	P	C43205H-B.ADA	P
C41203B-B.ADA	P	C43205I-B.ADA	P
C41204A.ADA	P	C43205J-B.ADA	P
C41205A-B.ADA	P	C43205K-B.ADA	P
C41206A.ADA	P	C43206A-B.ADA	P
C43001A-B.ADA	P	C43207A-B.ADA	P
B43002A-AB.ADA	P	C43207B-B.ADA	P
B43002B-AB.ADA	P	C43207C-B.ADA	P
C43003A-B.ADA	P	C43207D-B.ADA	P
C43003B-B.ADA	P	C43208A-B.ADA	P
C43003C-B.ADA	P	C43208B-B.ADA	P
C43003E-B.ADA	P	C43210A-B.ADA	P
C43003F-B.ADA	P	C43211A-B.ADA	P
C43003G-B.ADA	P	B43211B-B.ADA	P
C43003I-B.ADA	P	C43212A-B.ADA	P
C43003J-B.ADA	P	B43212B-B.ADA	P
C43003K-B.ADA	P	C43212C-B.ADA	P
C43003M-B.ADA	P	C43213A-B.ADA	P
C43003N-B.ADA	P	C43214A-B.ADA	P
C43003O-B.ADA	P	C43214B-B.ADA	P
C43003Q-B.ADA	P	C43214C-B.ADA	P
C43003R-B.ADA	P	C43214D-B.ADA	P
C43003S-B.ADA	P	C43214E-B.ADA	P
C43003U-B.ADA	P	C43214F-B.ADA	P
C43003V-B.ADA	P	C43215A-B.ADA	P
C43003W-B.ADA	P	C43215B-B.ADA	P
C43004A-B.ADA	P	B44001A-B.ADA	P
C43006A-B.ADA	P	B44002A-B.ADA	P
C43006B-B.ADA	P	B44002B-B.ADA	P
C43006C-B.ADA	P	B44002C.ADA	P
B44004A-B.ADA	P	C45101A.ADA	P
C42005A-B.ADA	P	C45101B.ADA	P
C42006A-B.ADA	P	C45101C.ADA	P
B43101A-B.ADA	P	C45101E.ADA	P
C43103A-B.ADA	P	C45101G-AB.ADA	P

C45101H-AB.ADA	P	C45241C-B.DEP	P
C45101I.ADA	P	C45241E-B.DEP	P
B45102A-AB.ADA	P	C45241F-B.DEP	N/A
C45103A-AB.ADA	P	C45241G-B.DEP	N/A
C45103B-AB.ADA	P	C45241H-B.DEP	N/A
C45103C-AB.ADA	P	C45241I-B.DEP	N/A
C45104A.ADA	P	C45241J-B.DEP	N/A
C45105A-AB.ADA	P	C45241K-B.DEP	N/A
C45105B-B.ADA	P	C45241L-B.DEP	N/A
C45106A.ADA	P	C45241M-B.DEP	N/A
C45201A.ADA	P	C45241N-B.DEP	N/A
C45201B.ADA	P	C45241O-B.DEP	N/A
C45202A-AB.ADA	P	C45241P-B.DEP	N/A
B45203A.ADA	P	C45241Q-B.DEP	N/A
B45203B-AB.ADA	P	C45241R-B.DEP	N/A
B45205A-AB.ADA	P	C45241S-B.DEP	N/A
B45206A-AB.ADA	P	C45241T-B.DEP	N/A
B45206B-B.ADA	P	C45241U-B.DEP	N/A
B45207A-AB.ADA	P	C45241V-B.DEP	N/A
B45207B-B.ADA	P	C45241W-B.DEP	N/A
B45207C-B.ADA	P	C45241X-B.DEP	N/A
B45207D-B.ADA	P	C45241Y-B.DEP	N/A
B45207G-B.ADA	P	B45261A-AB.ADA	P
B45207H-B.ADA	P	B45261B-AB.ADA	P
B45207I-B.ADA	P	B45261C-AB.ADA	P
B45207J-B.ADA	P	B45261D-AB.ADA	P
B45207M-AB.ADA	P	C45264A-B.ADA	P
B45207N-AB.ADA	P	C45274A-AB.ADA	P
B45207O-AB.ADA	P	C45274B-AB.ADA	P
B45207P-B.ADA	P	C45274C-AB.ADA	P
B45207S-AB.ADA	P	C45303A-B.ADA	P
B45207T-AB.ADA	P	C45321A-B.DEP	P
B45207U-AB.ADA	P	C45321B-B.DEP	P
B45207V-B.ADA	P	C45321C-B.DEP	P
B45208A-AB.ADA	P	C45321D-B.DEP	P
B45208B-B.ADA	P	C45321E-B.DEP	P
B45208C-B.ADA	P	C45321F-B.DEP	N/A
B45208G-AB.ADA	P	C45321G-B.DEP	N/A
B45208H-B.ADA	P	C45321H-B.DEP	N/A
B45208I-B.ADA	P	C45321I-B.DEP	N/A
B45208M-AB.ADA	P	C45321J-B.DEP	N/A
B45208N-AB.ADA	P	C45321K-B.DEP	N/A
B45208S-AB.ADA	P	C45321L-B.DEP	N/A
B45208T-AB.ADA	P	C45321M-B.DEP	N/A
C45210A.ADA	P	C45321N-B.DEP	N/A
C45220A.ADA	P	C45321O-B.DEP	N/A
C45220B.ADA	P	C45321P-B.DEP	N/A
C45220C.ADA	P	C45321Q-B.DEP	N/A
C45220D.ADA	P	C45321R-B.DEP	N/A
C45220E-B.ADA	P	C45321S-B.DEP	N/A
C45241A-B.DEP	P	C45321T-B.DEP	N/A
C45241B-B.DEP	P	C45321U-B.DEP	N/A
C45241C-B.DEP	P	C45321V-B.DEP	N/A

C45321W-B.DEP	N/A
C45321X-B.DEP	N/A
C45321Y-B.DEP	N/A
C45342A-AB.ADA	P
C45343A-AB.ADA	P
C45345A-AB.ADA	P
C45345B-AB.ADA	P
C45345C-AB.ADA	P
C45345D-AB.ADA	P
C45401A.ADA	P
C45401B-AB.ADA	P
B45402A.ADA	P
C45403A-B.ADA	P
C45421A-B.DEP	P
C45421B-B.DEP	P
C45421C-B.DEP	P
C45421D-B.DEP	P
C45421E-B.DEP	P
C45421F-B.DEP	N/A
C45421G-B.DEP	N/A
C45421H-B.DEP	N/A
C45421I-B.DEP	N/A
C45421J-B.DEP	N/A
C45421K-B.DEP	N/A
C45421L-B.DEP	N/A
C45421M-B.DEP	N/A
C45421N-B.DEP	N/A
C45421O-B.DEP	N/A
C45421P-B.DEP	N/A
C45421Q-B.DEP	N/A
C45421R-B.DEP	N/A
C45421S-B.DEP	N/A
C45421T-B.DEP	N/A
C45421U-B.DEP	N/A
C45421V-B.DEP	N/A
C45421W-B.DEP	N/A
C45421X-B.DEP	N/A
C45421Y-B.DEP	N/A
C45424A-B.DEP	P
C45424B-B.DEP	P
C45424C-B.DEP	P
C45424D-B.DEP	P
C45424E-B.DEP	P
C45424F-B.DEP	N/A
C45424G-B.DEP	N/A
C45424H-B.DEP	N/A
C45424I-B.DEP	N/A
C45424J-B.DEP	N/A
C45424K-B.DEP	N/A
C45424L-B.DEP	N/A
C45424M-B.DEP	N/A
C45424N-B.DEP	N/A
C45424O-B.DEP	N/A
C45424P-B.DEP	N/A
C45424Q-B.DEP	N/A
C45424R-B.DEP	N/A
C45424S-B.DEP	N/A
C45424T-B.DEP	N/A
C45424U-B.DEP	N/A
C45424V-B.DEP	N/A
C45424W-B.DEP	N/A
C45424X-B.DEP	N/A
C45424Y-B.DEP	N/A
C45505A-B.ADA	P
C45521A-B.DEP	W
C45521B-B.DEP	W
C45521C-B.DEP	W
C45521D-B.DEP	W
C45521E-B.DEP	W
C45521F-B.DEP	W
C45521G-B.DEP	W
C45521H-B.DEP	W
C45521I-B.DEP	W
C45521J-B.DEP	W
C45521K-B.DEP	W
C45521L-B.DEP	W
C45521M-B.DEP	W
C45521N-B.DEP	W
C45521O-B.DEP	W
C45521P-B.DEP	W
C45521Q-B.DEP	W
C45521R-B.DEP	W
C45521S-B.DEP	W
C45521T-B.DEP	W
C45521U-B.DEP	W
C45521V-B.DEP	W
C45521W-B.DEP	W
C45521X-B.DEP	W
C45521Y-B.DEP	W
B45522A.ADA	P
C45520A-B.ADA	P
B45530A-AB.ADA	P
C45021A.DEP	P
C45021B.DEP	P
C45021C.DEP	P
C45021D.DEP	P
C45021E.DEP	P
C45021F.DEP	N/A
C45621G.DEP	N/A
C45021H.DEP	N/A
C45621I.DEP	N/A
C45021J.DEP	N/A
C45021K.DEP	N/A
C45021L.DEP	N/A
C45021M.DEP	N/A
C45021N.DEP	N/A

C45621D.DEP	N/A
C45621P.DEP	N/A
C45621Q.DEP	N/A
C45621R.DEP	N/A
C45621S.DEP	N/A
C45621T.DEP	N/A
C45621U.DEP	N/A
C45621V.DEP	N/A
C45621W.DEP	N/A
C45621X.DEP	N/A
C45621Y.DEP	N/A
C45621Z.DEP	N/A
B48001A-B.ADA	N/A
B48001B-B.ADA	N/A
B48002A-B.ADA	N/A
B48002B-B.ADA	N/A
B48002C-B.ADA	N/A
B48002D-B.ADA	N/A
B48002E-B.ADA	N/A
B48002F-B.ADA	N/A
B48002G-B.ADA	N/A
B48003A-B.ADA	P
B48003B-B.ADA	P
B48003C-B.ADA	P
B48003D-B.ADA	P
B48003E-B.ADA	P
C48004A-B.ADA	P
C48004B-B.ADA	P
C48004C-B.ADA	P
C48004D-B.ADA	P
C48004E-B.ADA	P
C48004F-B.ADA	P
C48005A-B.ADA	P
C48005B-B.ADA	P
C48005C-B.ADA	W
C48006A-B.ADA	P
C48006B-B.ADA	W
C48007A-B.ADA	P
C48007B-B.ADA	P
C48007C-B.ADA	P
C48008A-B.ADA	P
C48008B-B.ADA	P
C48008C-B.ADA	P
C48008D-B.ADA	P
C48009A-B.ADA	P
C48009B-B.ADA	P
C48009C-B.ADA	P
C48009D-B.ADA	P
C48009E-B.ADA	P
C48009F-B.ADA	P
C48009G-B.ADA	P
C48009H-B.ADA	P
C48009I-B.ADA	P
C48009J-B.ADA	P
C48010A-B.ADA	P
C48012A-B.ADA	P
C4A001A.ADA	P
D4A002A-AB.ADA	P
D4A002B.ADA	P
C4A003A.ADA	P
D4A004A-AB.ADA	P
D4A004B.ADA	P
B4A006A-B.ADA	P
C4A010A-B.ADA	P
C4A011A.ADA	P
C4A013A.ADA	P
B4A016A.ADA	P

ACVC 1.6 test results chapter 5 IA..G - AVF

B51001A-A..ADA	P	C52103K-AB.ADA	P
C51002A-A..ADA	P	C52103L-AB.ADA	P
B51003A-AB.ADA	P	C52103M-AB.ADA	P
C51004A-B.ADA	P	C52103P-AB.ADA	P
B51004B-B.ADA	P	C52103R-AB.ADA	P
B51004C-B.ADA	P	C52103S-B.ADA	P
C52001A-B.ADA	P	C52103X-B.ADA	P
C52001B-AB.ADA	P	B52103Y-B.ADA	P
C52001C-AB.ADA	P	C52104A-AB.ADA	P
B52002A-B.ADA	P	C52104B-AB.ADA	P
B52002B-A..ADA	P	C52104C-AB.ADA	P
B52002C-AB.ADA	P	C52104F-AB.ADA	P
B52002D-AB.ADA	P	C52104G-AB.ADA	P
B52002E-AB.ADA	P	C52104H-AB.ADA	P
B52002F-B.ADA	P	C52104K-AB.ADA	P
B52002G-AB.ADA	P	C52104L-AB.ADA	P
B52003A-AB.ADA	P	C52104M-AB.ADA	P
B52003B-AB.ADA	P	C52104P-AB.ADA	P
B52003C-AB.ADA	P	C52104Q-AB.ADA	P
B52004A-B.ADA	P	C52104R-AB.ADA	P
B52004B-AB.ADA	P	C52104X-B.ADA	P
B52004C-AB.ADA	P	C52104Y-B.ADA	P
B52004D-AB.DEP	N/A	B53001A-AB.ADA	P
B52004E-AB.DEP	N/A	B53001B-AB.ADA	P
C52005A-AB.ADA	P	B53002A-AB.ADA	P
C52005B-AB.ADA	P	B53002B-AB.ADA	P
C52005C-AB.ADA	P	B53003A-AB.ADA	P
C52005D-AB.ADA	P	B53004A-AB.ADA	P
C52005E-AB.ADA	P	C53004B-B.ADA	P
C52005F-AB.ADA	P	C53005A-AB.ADA	P
B52006A-AB.ADA	P	C53005B-AB.ADA	P
C52007A-B.ADA	P	C53006A-AB.ADA	P
C52008A-AB.ADA	P	C53006B-AB.ADA	P
C52008B-B.ADA	P	C53007A-AB.ADA	P
C52009A-B.ADA	P	C53008A-AB.ADA	P
C52009B-B.ADA	P	B53009A-AB.ADA	P
C52010A-AB.ADA	P	B53009B-AB.ADA	P
C52011A-B.ADA	P	B53009C-AB.ADA	P
C52011B-AB.ADA	P	B54A01A-AB.ADA	P
C52012A-AB.ADA	P	B54A01B-AB.ADA	P
C52012B-AB.ADA	P	B54A01C-AB.ADA	P
C52013A-B.ADA	P	B54A01D-AB.ADA	P
C52101A-AB.ADA	P	B54A01E-AB.ADA	P
C52102A-AB.ADA	P	B54A01F-AB.ADA	P
C52102B-AB.ADA	P	B54A01G-AB.ADA	P
C52102C-AB.ADA	P	B54A01H-AB.ADA	P
C52102D-AB.ADA	P	B54A01I-AB.ADA	P
C52103A-AB.ADA	P	B54A01J-AB.ADA	P
C52103B-AB.ADA	P	B54A01K-AB.ADA	P
C52103C-AB.ADA	P	B54A01L-AB.ADA	P
C52103F-AB.ADA	P	C54A03A.ADA	P
C52103G-AB.ADA	P	C54AC4A-AB.ADA	P
C52103H-AB.ADA	P		

ACVC 1.6 test results chapter 5 IABG - AVF

B54A05A.ADA	P	B55A01T-AB.ADA	P
B54A05B.ADA	P	B55A01U-AB.ADA	P
C54A06A-AB.ADA	P	B55A01V-AB.ADA	P
C54A07A-AB.ADA	P	D55A03A-AB.ADA	P
B54A08A-B.ADA	P	D55A03B-AB.ADA	P
B54A20A.ADA	P	D55A03C-AB.ADA	P
B54A21A-B.ADA	P	D55A03D-AB.ADA	P
C54A22A-AB.ADA	P	D55A03E-AB.ADA	P
C54A23A-B.ADA	P	D55A03F-AB.ADA	P
C54A24A-AB.ADA	P	D55A03G-AB.ADA	P
C54A24B.ADA	P	D55A03H-AB.ADA	P
B54A25A-B.ADA	P	B55B01A-AB.ADA	P
C54A26A.ADA	P	B55BC1B-AB.ADA	P
C54A27A-AB.ADA	P	C55B03A-AB.ADA	P
B54A27B-AB.ADA	P	C55BC4A-AB.ADA	P
B54A27D-AB.ADA	P	C55B05A-AB.ADA	P
C54A41A.ADA	P	C55B06A-AB.ADA	P
C54A42A.ADA	P	C55B06B-AB.ADA	P
C54A42B.ADA	P	C55BC7A-AB.DEP	N/A
C54A42C.ADA	P	C55B07B-AB.DEP	N/A
C54A42D.ADA	P	C55B08A-B.ADA	P
C54A42E.ADA	P	C55B09A-AB.ADA	P
C54A42F.ADA	P	B55B09B-AB.ADA	P
C54A42G.ADA	P	B55B09C-AB.DEP	N/A
A54B01A-B.ADA	P	B55B09D-AB.DEP	N/A
B54B01B-B.TST	P	A55B12A-AB.ADA	P
B54B01C-B.ADA	P	A55B12B-B.ADA	P
A54B02A-B.ADA	P	B55B12C-AB.ADA	P
B54B02B-B.ADA	P	A55B13A-AB.ADA	P
B54B02C-B.ADA	P	A55B14A-AB.ADA	P
B54B02D-B.ADA	P	B55B14B-B.ADA	P
B54B04A-AB.ADA	P	C55B15A-B.ADA	P
B54B04B-AB.ADA	P	C55B16A-AB.DEP	P
B54B05A-AB.ADA	P	B55B16A-B.ADA	P
B55A01A-AB.ADA	P	C55C01A-B.ADA	P
B55A01B-AB.ADA	P	C55C02A-AB.ADA	P
B55A01C-AB.ADA	P	C55C02B-AB.ADA	P
B55A01D-AB.ADA	P	C55C03A-AB.ADA	P
B55A01E-AB.ADA	P	C55C03B-AB.ADA	P
B55A01F-AB.ADA	P	C55D01A-AB.ADA	P
B55A01G-AB.ADA	P	B56D01A-AB.ADA	P
B55A01H-AB.ADA	P	D56001B-AB.ADA	P
B55A01I-AB.ADA	P	B56001C-AB.ADA	P
B55A01J-AB.ADA	P	B56001D-AB.ADA	P
B55A01K-AB.ADA	P	B56001E-AB.ADA	P
B55A01L-AB.ADA	P	B56001F-AB.ADA	P
B55A01M-AB.ADA	P	B56001G-AB.ADA	P
B55A01N-AB.ADA	P	B56001H-AB.ADA	P
B55A01O-AB.ADA	P	C56002A-AB.ADA	P
B55A01P-AB.ADA	P	B57J01A-AB.ADA	P
B55A01Q-AB.ADA	P	B57001B-B.ADA	P
B55A01R-AB.ADA	P	B57001C-AB.ADA	P
B55A01S-AB.ADA	P	B57001D-AB.ADA	P

ACVC 1.6 test results chapter 5 I G - AVF

C37002A-AB.ADA P
C37003A-AB.ADA P
C37004A-AB.ADA P
C37004B-AB.ADA P
C37004C-AB.ADA P
C37005A-B.ADA P
B38001A-AB.ADA P
B38002A-B.ADA P
B38002B-AB.ADA P
B38002C-AB.ADA P
B38003A-B.ADA P
B38003B-AB.ADA P
C38004A-AB.ADA P
C38004B-AB.ADA P
C38004C-AB.ADA P
C38004D-B.ADA P
C38004F-AB.ADA P
C38004G-AB.ADA P
C38005A-AB.ADA P
C38005B-AB.ADA P
C38005M-AB.ADA P
C38006A-AB.ADA P
C38006B-AB.ADA P
B39001A-AB.ADA P
C39001B-AB.ADA P
B39001C-AB.ADA P
B39001D-AB.ADA P
B39001E-AB.ADA P
B39001F-AB.ADA P
B39001G-AB.ADA P
B39001M-AB.ADA P
B39001I-AB.ADA P
C39002A-AB.ADA P
C39002B-AB.ADA P
C39002C-B.ADA P

ACVC 1.6 test results chapter 6 IABG - AVF

Be1C01A-AB.ADA	P	B63009C0-B.ADA	P
Be1C01B-AB.ADA	P	B63009C1-B.ADA	P
Be1C01C-AB.ADA	P	B63009C2-B.ADA	P
Be1C01D-AB.ADA	P	B63009C3M-B.ADA	P
Be1C01E-AB.ADA	P	B63010A-AB.ADA	P
Be1C01F-AB.ADA	P	B63102A-B.ADA	P
Be1C01G-AB.ADA	P	B63103A-B.ADA	P
Be1C01H-AB.ADA	P	A63202A-AB.ADA	P
Be1C01I-AB.ADA	P	B64001A-B.ADA	P
Be1C01J-AB.ADA	P	B64002A-B.ADA	P
Be1C01K-AB.ADA	P	C64002B-B.ADA	P
Be1C01L-AB.ADA	P	B64002C-B.ADA	P
Be1C01M-AB.ADA	P	B64003A-B.ADA	P
Be1C01N-AB.ADA	P	B64004A-B.ADA	P
Be1C01O-AB.ADA	P	B64004B-B.ADA	P
Be1C01P-AB.ADA	P	B64004C-B.ADA	P
Be1C01Q-AB.ADA	P	B64004D-B.ADA	P
Be1C01R-AB.ADA	P	B64004E-B.ADA	P
Be1C01S-AB.ADA	P	B64004F-B.ADA	P
Be1C01T-AB.ADA	P	C64004G-B.ADA	P
Be1C01U-AB.ADA	P	C64005A-B.ADA	P
Be1C01V-AB.ADA	P	C64005B-B.ADA	P
Be1C01W-AB.ADA	P	C64005C-B.ADA	P
Be1C03A-AB.ADA	P	C64005DOM-B.ADA	P
Ce1C03B-AB.ADA	P	C64005DA-B.ADA	P
Be1C04A-B.ADA	P	C64005DB-B.ADA	P
Ce1C05A-B.ADA	P	C64005DC-B.ADA	P
Ce1C06A-B.ADA	P	D64005EOM-B.ADA	P
Ce1C10A-AB.ADA	P	D64005EA-B.ADA	P
Be1C11A-B.ADA	P	D64005EB-B.ADA	P
Be1C12A-B.ADA	P	D64005EC-B.ADA	P
Be2001A-AB.ADA	P	D64005ED-B.ADA	P
Be2C01B-AB.ADA	P	D64005EE-B.ADA	P
Be2001C-AB.ADA	P	D64005EF-B.ADA	P
Be2C01D-AB.ADA	P	D64005FOM-B.ADA	P
Ce2002A-B.ADA	P	D64005FA-B.ADA	P
Ce2003A-B.ADA	P	D64005FB-B.ADA	P
Ce2003B-B.ADA	P	D64005FC-B.ADA	P
Ce2004A-AB.ADA	P	D64005FD-B.ADA	P
Ce2005A-B.ADA	P	D64005FE-B.ADA	P
Be2005B-B.ADA	P	D64005FF-B.ADA	P
Be2005C-B.ADA	P	D64005FG-B.ADA	P
AE2005D-B.ADA	P	D64005FH-B.ADA	P
BE2005E-B.ADA	P	D64005FI-B.ADA	P
Be2005F-B.ADA	P	D64005FJ-B.ADA	P
Be3001A-AB.ADA	P	D64005GOM-B.ADA	P
BE3001B-AB.ADA	P	D64005GA-B.ADA	P
Ce3004A-AB.ADA	P	D64005GB-B.ADA	P
Be3005A-AB.ADA	P	D64005GC-B.ADA	P
B63005B-AB.ADA	P	D64005GD-B.ADA	P
Be3005C-AB.ADA	P	D64005GE-B.ADA	P
Be3009A-B.ADA	P	D64005GF-B.ADA	P
Be3009B-B.ADA	P	D64005GG-B.ADA	P

ACVC 1.0 test results chapter 6 IABG - AVF

D64005GM-B.ADA	P
D64005GI-B.ADA	P
D64005GJ-B.ADA	P
D64005GK-B.ADA	P
D64005GL-B.ADA	P
D64005GM-B.ADA	P
D64005GN-B.ADA	P
D64005GO-B.ADA	P
D64005GP-B.ADA	P
D64005GQ-B.ADA	P
E64005CA-B.ADA	P
B64101A-B.ADA	P
C64103A-B.ADA	P
C64103B-B.ADA	P
C64103C-B.ADA	W
C64103D-B.ADA	W
C64103E-B.ADA	P
C64103F-B.ADA	P
C64104A-AB.ADA	P
C64104B-AB.ADA	P
C64104C-AB.ADA	P
C64104D-AB.ADA	P
C64104E-AB.ADA	P
C64104F-AB.ADA	P
C64104G-AB.ADA	P
C64104H-B.ADA	P
C64104I-B.ADA	P
C64104J-B.ADA	P
C64104K-AB.ADA	P
C64104L-AB.ADA	P
C64104M-AB.ADA	P
C64104N-B.ADA	P
C64104O-B.ADA	P
C64105A-AB.ADA	P
C64105B-AB.ADA	P
C64105C-AB.ADA	P
C64105D-AB.ADA	P
C64105E-AB.ADA	W
C64105F-AB.ADA	W
C64106A-B.ADA	P
C64106B-B.ADA	P
C64106C-B.ADA	P
C64106D-B.ADA	P
C64107A-B.ADA	P
C64108A-B.ADA	P
B64201A-B.ADA	P
C64201B-B.ADA	P
C64201C-B.ADA	P
C64202A-B.ADA	P
B65001A-B.ADA	P
B65002A-AB.ADA	P
B65002B-AB.ADA	P
C65003A-B.ADA	P
C65003B-B.ADA	P
B66001A-B.ADA	W
B66001B-B.ADA	P
B66001C-B.ADA	P
C66002A-B.ADA	P
C66002C-AB.ADA	P
C66002D-AB.ADA	P
C66002E-AB.ADA	P
C66002F-AB.ADA	P
C66002G-B.ADA	P
B67001A-B.ADA	W
B67001B-B.ADA	P
B67001C-B.ADA	P
B67001D-B.ADA	P
B67001E-B.ADA	P
B67001F-B.ADA	P
B67001G-B.ADA	P
C67002A-B.ADA	P
C67002B-B.ADA	P
C67002C-B.ADA	P
C67002D-B.ADA	P
C67002E-B.ADA	P
C67003A-B.ADA	P
C67003B-B.ADA	P
C67003C-AB.ADA	P
C67003D-B.ADA	P
C67003E-AB.ADA	P
B67004A-B.ADA	W
C67005A-B.ADA	P
C67005B-B.ADA	P
C67005C-B.ADA	P
C67005D-B.ADA	P

B71001A-AB.ADA	P
B71001B-AB.ADA	P
B71001C-AB.ADA	P
B71001D-AB.ADA	P
B71001E-AB.ADA	P
B71001F-AB.ADA	P
B71001G-AB.ADA	P
B71001H-AB.ADA	P
B71001I-AB.ADA	P
B71001J-AB.ADA	P
B71001K-AB.ADA	P
B71001L-AB.ADA	P
B71001M-AB.ADA	P
B71001N-AB.ADA	P
B71001O-AB.ADA	P
B71001P-AB.ADA	P
B71001Q-AB.ADA	P
B71001R-AB.ADA	P
B71001T-AB.ADA	P
B71001U-AB.ADA	P
B71001V-AB.ADA	P
B71001W-AB.ADA	P
A71002A-AB.ADA	P
B71002B-AB.ADA	P
A71004A-AB.ADA	P
A72001A-AB.ADA	P
C72001B-AB.ADA	P
B73001A-AB.ADA	P
B73001B-AB.ADA	P
B73001C-B.ADA	P
B73001D-B.ADA	P
B73001E-AB.ADA	P
B73001F-AB.ADA	P
B73001G-B.ADA	P
B73001H-B.ADA	P
A73001I-AB.ADA	P
A73001J-AB.ADA	P
C73002A-B.ADA	P
B73004A-AB.ADA	P
B74001A-AB.ADA	P
B74001B-AB.ADA	P
B74003A-B.ADA	P
A74004A-AB.ADA	P
B74101A-B.ADA	P
B74103A-B.ADA	P
B74103B-B.ADA	P
B74103C-B.ADA	P
B74103D-B.ADA	P
B74103E-B.ADA	P
B74103F-B.ADA	W
B74103G-B.ADA	P
B74104A-B.ADA	P
B74105A-B.ADA	P
A74105B-B.ADA	P
B74105C-B.ADA	P
A74106A-AB.ADA	P
A74106B-AB.ADA	P
A74106C-AB.ADA	P
B74201A-AB.ADA	P
B74201A-B.ADA	P
B74201B-B.ADA	P
A74201E-B.ADA	P
A74201F-B.ADA	P
C74201A-B.ADA	P
C74201B-B.ADA	P
B74207A-B.ADA	W
C74207B-B.ADA	P
C74209A-AB.ADA	P
C74210A-AB.ADA	P
C74211A-B.ADA	P
C74211B-B.ADA	P
B74301A-B.ADA	P
C74302A-B.ADA	P
B74304A-B.ADA	P
B74304B-B.ADA	P
B74304C-B.ADA	P
C74305A-B.ADA	P
C74305B-B.ADA	P
B74401A-B.ADA	P
B74401B-B.ADA	P
C74402A-B.ADA	P
C74402B-B.ADA	P
B74409A-B.ADA	P
C74409B-B.ADA	P

ACVC 1.8 test results chapter 8 [illegible] - AVF

B83A01A-AB.ADA	P	B85007C-B.ADA	P
B83A01B-B.ADA	P	B85007D-B.ADA	P
B83A01C.ADA	P	C85007E-B.ADA	P
A83A02A.ADA	P	B85012A-B.ADA	P
AB3A02B.ADA	P	C85013A-B.ADA	P
B83A05A-AB.ADA	P	A85013B-B.ADA	P
AB3A06A-B.ADA	P	B85013C-B.ADA	P
B83A06B-B.ADA	P	B85015A-B.ADA	P
BB3A06H-B.ADA	P	B86001A0-AB.ADA	P
B83B01A-AB.ADA	P	B86001A1M-AB.ADA	P
CB3B02A.ADA	P	B86001B0M-B.ADA	P
CB3B02B.ADA	P	B86001BA-B.ADA	P
BB3B02C.ADA	P	B86001BB-B.ADA	P
B83C01A-AB.ADA	P	B86001BC-B.ADA	P
CB3C01B.ADA	P	B86001BD-B.ADA	P
AB3C01C.ADA	P	B86001BE-B.ADA	P
AB3C01D.ADA	P	B86001BF-B.ADA	P
AB3C01E.ADA	P	B86001BG-B.ADA	P
AB3C01F.ADA	P	B86001BH-B.ADA	P
AB3C01G.ADA	P	B86001BI-B.ADA	P
AB3C01H.ADA	P	B86001BJ-B.ADA	P
AB3C01I.ADA	P	B86001BK-B.ADA	P
AB3C01J.ADA	P	B86001BL-B.ADA	P
BB3C02A.ADA	P	B86001BM-B.ADA	P
CB3C02A.ADA	P	B86001BO-B.ADA	P
CB3C02B.ADA	P	B86001BU-B.ADA	P
BB3C02C-B.ADA	P	B86001BV-B.ADA	P
CB3C03A.ADA	P	B86001BW-B.ADA	P
CB3C04A.ADA	P	B86001BX-B.ADA	P
CB3F01A.ADA	P	B86001C0M-AB.DEP	N/A
CB3F01B.ADA	P	B86001CP-AB.DEP	N/A
CB3F01C0.ADA	P	B86001CQ-AB.DEP	N/A
CB3F01C1.ADA	P	B86001CR-AB.DEP	N/A
CB3F01C2M.ADA	P	B86001CS-AB.DEP	N/A
CB3F01D0M.ADA	P	B86001D0M-AB.TST	P
CB3F01D1.ADA	P	B86001DT-AB.TST	N/A
BB3F02A.ADA	P	CB6001E-B.ADA	P
BB3F02B.ADA	P	CB6001F-B.DEP	N/A
CB3F03A.ADA	P	CB6002A0.ADA	P
CB3F03B.ADA	P	CB6002A1.ADA	P
CB3F03C0.ADA	P	CB6002A2M.ADA	P
CB3F03C1.ADA	P	CB6002B1.ADA	P
CB3F03C2M.ADA	P	CB6002B2M.ADA	P
CB3F03D0M.ADA	P	CB6003A-B.ADA	P
CB3F03D1.ADA	P	CB7AC5A-B.ADA	P
BB3F04A-AB.ADA	P	CB7A05B-B.ADA	P
BB4UC1A-AB.ADA	P	CB7B02A-B.ADA	P
CB4U02A-B.ADA	P	CB7BC2B-B.ADA	P
BB4U02B-B.ADA	P	CB7B03A-B.ADA	P
BB4U04A-B.ADA	P	CB7BC4A-B.ADA	P
BB4U06A-B.ADA	P	CB7BC4B-B.ADA	P
CB5007A-B.ADA	P	CB7B04C-B.ADA	P
BB5U07B-B.ADA	P	CB7B05A-B.ADA	P

ACVC 1.9 test results chapter 8 IA06 - AVF

CE7A05A-B.ADA	P	CE7A43A-B.ADA	P
CE7A07A-B.ADA	P	CE7A44A-B.ADA	P
CE7A07B-B.ADA	P	CE7A45A-B.ADA	P
CE7A07C-B.ADA	P	CE7B45C-B.ADA	P
CE7A07D-B.ADA	P	CE7B47A-B.ADA	P
CE7A07E-B.ADA	P	CE7A48A-B.ADA	P
CE7A08A-B.ADA	P	CE7A48B-B.ADA	P
CE7A09A-B.ADA	P	CE7B48C-B.ADA	P
CE7A09B-B.ADA	P	CE7B54A-B.ADA	P
CE7A09C-B.ADA	P	CE7B57A-B.ADA	P
CE7A10A-B.ADA	P	CE7B62A-B.DEP	P
CE7B11A-B.ADA	P	CE7B62B-B.DEP	P
CE7B11B-B.ADA	P	CE7B62C-B.DEP	P
CE7B13A-B.ADA	P		
CE7A14A-B.ADA	P		
CE7B14B-B.ADA	P		
CE7A14C-B.ADA	P		
CE7A14D-B.ADA	P		
CE7A15A-B.ADA	P		
CE7A16A-B.ADA	P		
CE7A17A-B.ADA	P		
CE7A18A-B.ADA	P		
CE7A19A-B.ADA	P		
CE7A23A-B.ADA	P		
CE7A23B-B.ADA	P		
CE7A24A-B.ADA	P		
CE7B24B-B.ADA	P		
CE7A26B-B.ADA	P		
CE7B27A-B.ADA	P		
CE7A28A-B.ADA	P		
CE7A29A-B.ADA	P		
CE7A30A-B.ADA	P		
CE7A31A-B.ADA	P		
CE7A32A-B.ADA	P		
CE7A33A-B.ADA	P		
CE7A34A-B.ADA	P		
CE7B34B-B.ADA	P		
CE7A34C-B.ADA	P		
CE7A35A-B.ADA	P		
CE7A35B-B.ADA	P		
CE7B35C-B.ADA	P		
CE7A37A-B.ADA	P		
CE7A37B-B.ADA	P		
CE7A37C-B.ADA	P		
CE7A37D-B.ADA	P		
CE7B37E-B.ADA	P		
CE7B37F-B.ADA	P		
CE7E38A-B.ADA	P		
CE7A39A-B.ADA	P		
CE7A40A-B.ADA	P		
CE7A41A-B.ADA	P		
CE7A42A-B.ADA	P		

ACVC 1.6 test results chapter 9 IABG - AVF

C96006A-B.ADA	P	C940C1A-B.ADA	P
B91001A-AB.ADA	P	C940C2A-B.ADA	P
B91001B-AB.ADA	P	C940C2B-B.ADA	P
B91001C-AB.ADA	P	C94003A-B.ADA	P
B91001D-AB.ADA	P	C94004A-B.ADA	P
B91001E-AB.ADA	P	C94004B-B.ADA	P
B91001F-AB.ADA	P	C94004C-B.ADA	P
B91001G-B.ADA	P	C94005A-B.ADA	P
B91002A-B.ADA	P	C94005B-B.ADA	P
B91002B-B.ADA	P	C94006A-B.ADA	P
B91002C-B.ADA	P	C94007A-B.ADA	P
B91002D-B.ADA	P	C94007B-B.ADA	P
B91002E-B.ADA	P	C94020A-B.ADA	P
B91002F-B.ADA	P	C94021A-B.ADA	P
B91002G-B.ADA	P	C94021B-B.ADA	P
B91002H-B.ADA	P	C940ACA-B.ADA	P
B91002I-B.ADA	P	C940ACB-B.ADA	P
B91002J-B.ADA	P	C940ADA-B.ADA	P
B91002K-B.ADA	P	C940AGA-B.ADA	P
B91002L-B.ADA	P	C940AGB-B.ADA	P
A91002M-B.ADA	P	C940AHA-B.ADA	P
B91003A-AB.ADA	P	C940AIA-B.ADA	P
B91004A-B.ADA	P	C940BAA-B.ADA	P
B91004B-B.ADA	P	C940BBA-B.ADA	P
B91004C-B.ADA	P	B95001A.ADA	P
B91004D-AB.ADA	P	B95001B-AB.ADA	P
C91004A.ADA	P	B95002A.ADA	P
B91004E-B.ADA	P	B95004A-AB.ADA	P
C91004B-B.ADA	P	B95004B-AB.ADA	P
C91004C-B.ADA	P	A95005A.ADA	P
C91004D-B.ADA	P	B95006A.ADA	P
C92002A.ADA	P	B95006B-AB.ADA	P
C92003A.ADA	P	B95006C-AB.ADA	P
B92004A-B.ADA	P	B95006D-AB.ADA	P
C92004A-B.ADA	P	B95007A-AB.ADA	P
C92004B-B.ADA	P	B95007B-AB.ADA	P
C92004C-B.ADA	P	C95008A-AB.ADA	P
B92004D-B.ADA	P	C95009A-B.ADA	P
C92004I-B.ADA	P	C95009B.ADA	P
B92004J-B.ADA	P	C95010A.ADA	P
C93001A-B.ADA	P	C95011A.ADA	P
C93002A-B.ADA	P	C95012A-B.ADA	P
C93003A-B.ADA	P	C95013A-B.ADA	P
C93005A-B.ADA	W	B95020A-B.ADA	P
C93005B-B.ADA	W	B95020B-B.ADA	P
C93005C-B.ADA	W	B95020M-B.ADA	P
C93006A-AB.ADA	P	B95020N-B.ADA	P
C93007B-B.ADA	W	C95021A-B.ADA	P
C93008A-B.ADA	W	C95022A-B.ADA	P
C93008B-B.ADA	P	C95022B-B.ADA	P
C93008F-B.ADA	P	B95023A-B.ADA	P
C93008J-B.ADA	P	B95023B-B.ADA	P
C93008A-B.ADA	P	B95024A-B.ADA	P

C95UACB-B.ADA	P
B95UADA-B.ADA	P
B95UAFA-B.ADA	P
B95UAHA-B.ADA	P
B95UAJA-B.ADA	P
B95UBAA-B.ADA	P
C95UBGA-B.ADA	P
C95UBHA-B.ADA	P
C95UBJA-B.ADA	P
C95UCAA-B.ADA	P
C95UCBA-B.ADA	P
C95UCHA-B.ADA	P
C95UCHC-B.ADA	P
C95UDEA-B.ADA	P
C95UDEB-B.ADA	P
C95UDGA-B.ADA	P
B95UDHA-B.ADA	P
C96001A-B.ADA	P
B95UGZA-B.ADA	P
B96003A-B.ADA	P
C96004A-B.ADA	P
C96005A-B.ADA	P
C96005B-B.TST	N/A
C96005C-B.TST	P
C96005D-B.ADA	P
C96005E-B.ADA	P
C96006A-B.ADA	P
C96007A-B.ADA	P
C96008A-B.ADA	P
C96008B-B.ADA	P
B97101A-AB.ADA	P
B97101B-AB.ADA	P
B97101C-AB.ADA	P
B97101D-AB.ADA	P
B97101E-AB.ADA	P
B97102A-AB.ADA	P
B97102B-AB.ADA	P
B97102C-AB.ADA	P
B97102D-AB.ADA	P
B97102E-AB.ADA	P
B97102F-AB.ADA	P
B97102G-AB.ADA	P
B97102H-AB.ADA	P
B97102I-AB.ADA	P
B97103A-AB.ADA	P
B97103B-AB.ADA	P
B97103D-AB.ADA	P
B97103E-AB.ADA	P
B97104A-AB.ADA	P
B97104B-AB.ADA	P
B97104C-AB.ADA	P
B97104D-AB.ADA	P
B97104E-AB.ADA	P
B97104F-AB.ADA	P
B97104G-AB.ADA	P
B97104O-AB.ADA	P
B97107A-AB.ADA	P
B97108A-AB.ADA	P
B97108B-AB.ADA	P
B97109A-AB.ADA	P
B97110A-AB.ADA	P
B97110B-AB.ADA	P
B97111A-AB.ADA	P
C97113A-B.ADA	P
C97114A-B.ADA	P
C97115A-B.ADA	P
C97201A-AB.ADA	P
C97201D-AB.ADA	P
C97201E-AB.ADA	P
C97201G-AB.ADA	P
C97201H-AB.ADA	P
C97201X-AB.ADA	P
C97202A-AB.ADA	P
C97203A-AB.ADA	P
C97203B-AB.ADA	P
C97204A-B.ADA	P
C97303A-AB.ADA	P
C97303B-AB.ADA	P
C97304A-B.ADA	P
B99001A-AB.ADA	P
B99001B-B.ADA	P
B99001C-B.ADA	P
B99002B-B.ADA	P
B99002C-B.ADA	P
B99003A-AB.ADA	P
B9A001A-AB.ADA	P
B9A001B-AB.ADA	P
C9A003A-B.ADA	P
C9A004A-B.ADA	P
C9A005A-B.ADA	P
C9A006A-B.ADA	P
C9A007A-B.ADA	P
C9A009A-B.ADA	P
C9A009B-B.ADA	P
C9A009C-B.ADA	P
C9A009D-B.ADA	P
C9A009E-B.ADA	P
C9A009F-B.ADA	P
C9A009G-B.ADA	P
C9A009H-B.ADA	P

ACVC 1.6 test results chapter 10 IAEG - AVF

CA1002A0-B.ADA	P	CA1012A4M-B.DEP	P
CA1002A1-B.ADA	P	CA1012B0-B.ADA	P
CA1002A2-B.ADA	P	CA1012B2-B.ADA	P
CA1002A3M-B.ADA	P	CA1012B4M-B.ADA	P
CA1002A4-B.ADA	P	CA1013A0-B.ADA	P
CA1002A5-B.ADA	P	CA1013A1-B.ADA	P
CA1002A6-B.ADA	P	CA1013A2-B.ADA	P
CA1002A7-B.ADA	P	CA1013A3-B.ADA	P
CA1002A8-B.ADA	P	CA1013A4-B.ADA	P
CA1002A9-B.ADA	P	CA1013A5-B.ADA	P
CA1003A-AB.ADA	P	CA1013A6M-B.ADA	P
CA1003B-AB.ADA	W	CA1014A0M-AB.ADA	P
CA1004A-AB.ADA	P	CA1014A1-AB.ADA	P
CA1005A-AB.ADA	P	CA1014A2-AB.ADA	P
CA1006A-AB.ADA	P	CA1014A3-AB.ADA	P
CA1007A0-AB.ADA	P	BA1102CA0M-B.ADA	P
CA1007A1M-AB.ADA	P	BA1102CA1-B.ADA	P
CA1008A0-AB.ADA	W	BA1102CA2-B.ADA	P
CA1008A1M-AB.ADA	W	BA1102CA3-B.ADA	P
CA1009A0-AB.ADA	P	BA1102CA4-B.ADA	P
CA1009A1-AB.ADA	P	BA1102CA5-B.ADA	P
CA1009A2-AB.ADA	P	BA1102CA6-B.ADA	P
CA1009A3-AB.ADA	P	BA1102CA7-B.ADA	P
CA1009A4M-AB.ADA	P	BA1102CA8-B.ADA	P
CA1011A0-B.ADA	W	BA1102CB0-B.ADA	P
CA1011A1-B.ADA	W	BA1102CB1-B.ADA	P
CA1011A2-B.ADA	W	BA1102CB2-B.ADA	P
CA1011A3-B.ADA	W	BA1102CB3-B.ADA	P
CA1011A4-B.ADA	W	BA1102CB4-B.ADA	P
CA1011A5-B.ADA	W	BA1102CB5-B.ADA	P
CA1011A6M-B.ADA	W	BA1102CB6M-B.ADA	P
BA1011B0M-B.ADA	P	BA1102CC0M-B.ADA	P
BA1011B1-B.ADA	P	BA1102CC1-B.ADA	P
BA1011B2-B.ADA	P	BA1102CC2-B.ADA	P
BA1011B3-B.ADA	P	BA1102CC3-B.ADA	P
BA1011B4-B.ADA	P	BA1102CC4-B.ADA	P
BA1011B5-B.ADA	P	BA1102CC5-B.ADA	P
BA1011B6-B.ADA	P	CA1022A0-B.ADA	P
BA1011B7-B.ADA	P	CA1022A1-B.ADA	P
BA1011B8-B.ADA	P	CA1022A2-B.ADA	P
BA1011C0M-B.ADA	P	CA1022A3-B.ADA	P
BA1011C1-B.ADA	P	CA1022A4-B.ADA	P
BA1011C2-B.ADA	P	CA1022A5-B.ADA	P
BA1011C3-B.ADA	P	CA1022A6M-B.ADA	P
BA1011C4-B.ADA	P	BA1101A-AB.ADA	P
BA1011C5-B.ADA	P	BA1101B0M-B.ADA	P
BA1011C6-B.ADA	P	BA1101B1-B.ADA	P
BA1011C7-B.ADA	P	BA1101B2-B.ADA	P
BA1011C8-B.ADA	P	BA1101B3-B.ADA	P
CA1012A0-B.DEP	P	BA1101B4-B.ADA	P
CA1012A1-B.DEP	P	BA1101C0-B.ADA	P
CA1012A2-B.DEP	P	BA1101C1-B.ADA	P
CA1012A3-B.DEP	P	BA1101C2M-B.ADA	P

BA1101C3-B.ADA P
BA1101C4-B.ADA P
BA1101C5-B.ADA P
BA1101D-AB.ADA P
BA1101E-B.ADA P
BA1101F-B.ADA P
BA1101G-B.ADA P
BA1101H0-B.ADA P
BA1101H1M-B.ADA P
CA1102A0-B.ADA P
CA1102A1-B.ADA P
CA1102A2M-B.ADA P
CA1105A0-B.ADA P
CA1105A1M-B.ADA P
CA1105B0-B.ADA P
CA1105B1-B.ADA P
CA1105B2-B.ADA P
CA1105B3M-B.ADA P
CA1105B4-B.ADA P
CA1105B5-B.ADA P
CA1107A0.ADA P
CA1107A1M.ADA P
CA1107A2.ADA P
CA1108A-B.ADA W
CA1108B-B.ADA W
BA2001A-AB.ADA P
BA2001B-AB.ADA P
BA2001C-AB.ADA P
BA2001D-AB.ADA P
BA2001E0M-AB.ADA W
BA2001E1-AB.ADA W
BA2001E2-AB.ADA W
BA2001F0M-AB.ADA P
BA2001F1-AB.ADA P
BA2001F2-AB.ADA P
BA2001G0M-AB.ADA P
BA2001G1-AB.ADA P
CA2001H0-B.ADA P
CA2001H1-B.ADA P
CA2001H2-B.ADA P
CA2001H3M-B.ADA P
CA2002A0M-B.ADA P
CA2002A1-B.ADA P
CA2002A2-B.ADA P
CA2003A0M-AB.ADA P
CA2003A1-BB.ADA P
BA2003B0M-AB.ADA P
BA2003B1-AB.ADA P
CA2004A0M-AB.ADA P
CA2004A1-AB.ADA P
CA2004A2-AB.ADA P
CA2004A3-AB.ADA P
CA2004A4-AB.ADA P

CA2007A0M-AB.ADA P
CA2007A1-AB.ADA P
CA2007A2-AB.ADA P
CA2007A3-AB.ADA P
CA2008A0M-B.ADA P
CA2008A1-B.ADA P
CA2008A2-B.ADA P
CA2009A-B.DEP P
CA2009B-B.DEP W
CA2009C0M-B.DEP P
CA2009C1-B.DEP P
CA2009D-B.DEP P
CA2009E-B.DEP W
CA2009F0M-B.DEP W
CA2009F1-B.DEP W
BA2013A-B.ADA P
BA2013B-B.ADA P
BA3001A0M-AB.ADA P
BA3001A1-AB.ADA P
BA3001A2-AB.ADA P
BA3001A3-AB.ADA P
BA3001B0M.ADA P
BA3001B1.ADA P
BA3001C0M-AB.ADA P
BA3001C1-AB.ADA P
BA3001D0M-AB.ADA P
BA3001D1-AB.ADA P
BA3001E0M-AB.ADA P
BA3001E1-AB.ADA P
BA3001E2-AB.ADA P
BA3001E3-AB.ADA P
BA3001F0M-AB.ADA P
BA3001F1-AB.ADA P
BA3001F2-AB.ADA P
BA3001F3-AB.ADA P
CA3002A0-B.ADA P
CA3002A1-B.ADA P
CA3002A2M-B.ADA P
CA3002A3-B.ADA P
LA3004A0-AB.ADA P
LA3004A1-AB.ADA P
LA3004A2-AB.ADA P
LA3004A3-AB.ADA P
LA3004A4-AB.ADA P
LA3004A5-AB.ADA P
LA3004A6M-AB.ADA P
LA3004B0-B.ADA P
LA3004B1-B.ADA P
LA3004B2-B.ADA P
LA3004B3-B.ADA P
LA3004B4-B.ADA P
LA3004B5-B.ADA P
LA3004B6M-B.ADA P

BA3006A0-B.ADA	P
BA3006A1-B.ADA	P
BA3006A2-B.ADA	P
BA3006A3-B.ADA	P
BA3006A4-B.ADA	P
BA3006A5-B.ADA	P
BA3006A6M-B.ADA	P
BA3006B0-B.ADA	P
BA3006B1-B.ADA	P
BA3006B2-B.ADA	P
BA3006B3-B.ADA	P
BA3006B4M-B.ADA	P
CA3006C0-B.ADA	P
CA3006C1-B.ADA	P
CA3006C2-B.ADA	P
CA3006C3-B.ADA	P
CA3006C4-B.ADA	P
CA3006C5M-B.ADA	P
CA3006D0-B.ADA	P
CA3006D1-B.ADA	P
CA3006D2-B.ADA	P
CA3006D3M-B.ADA	P
CA3006E0-B.ADA	P
CA3006E1-B.ADA	P
CA3006E2-B.ADA	P
CA3006E3-B.ADA	P
CA3006E4-B.ADA	P
CA3006E5-B.ADA	P
CA3006E6M-B.ADA	P
BA3007A0-B.ADA	P
BA3007A1-B.ADA	P
BA3007A2-B.ADA	P
BA3007A3-B.ADA	P
BA3007A4-B.ADA	P
BA3007A5M-B.ADA	P
BA3007B0-B.ADA	P
BA3007B1-B.ADA	P
BA3007B2-B.ADA	P
BA3007B3-B.ADA	P
BA3007B4-B.ADA	P
BA3007B5-B.ADA	P
BA3007B6-B.ADA	P
BA3007B7-B.ADA	P
BA3007B8M-B.ADA	P
BA3008A0-B.ADA	P
BA3008A1-B.ADA	P
BA3008A2-B.ADA	P
BA3008A3-B.ADA	P
BA3008A4-B.ADA	P
BA3008A5M-B.ADA	P
BA3008B0-B.ADA	P
BA3008B1-B.ADA	P
BA3008B2-B.ADA	P
BA3008B3-B.ADA	P
BA3008B4-B.ADA	P
BA3008B5-B.ADA	P
BA3008B6M-B.ADA	P
BA3013A0-B.ADA	P
BA3013A1-B.ADA	P
BA3013A2-B.ADA	P
BA3013A3-B.ADA	P
BA3013A4-B.ADA	P
BA3013A5-B.ADA	P
BA3013A6-B.ADA	P
BA3013A7M-B.ADA	P
LA5001A0-B.ADA	P
LA5001A1-B.ADA	P
LA5001A2-B.ADA	P
LA5001A3-B.ADA	P
LA5001A4-B.ADA	P
LA5001A5-B.ADA	P
LA5001A6-B.ADA	P
LA5001A7M-B.ADA	P
CA5002A-B.ADA	P
CA5002B0-B.ADA	P
CA5002B1-B.ADA	P
CA5002B2-B.ADA	P
CA5002B3-B.ADA	P
CA5002B4-B.ADA	P
CA5002B5-B.ADA	P
CA5002B6-B.ADA	P
CA5002B7M-B.ADA	P
CA5003A0-B.ADA	P
CA5003A1-B.ADA	P
CA5003A2-B.ADA	P
CA5003A3-B.ADA	P
CA5003A4-B.ADA	P
CA5003A5-B.ADA	P
CA5003A6M-B.ADA	P
CA5003B0-B.ADA	P
CA5003B1-B.ADA	P
CA5003B2-B.ADA	P
CA5003B3-B.ADA	P
CA5003B4-B.ADA	P
CA5003B5M-B.ADA	P
CA5004A-B.ADA	P
CA5004B-B.ADA	P

ACVC 1.6 test results chapter 11 IABO - AVF

CB1001A-B.ADA	P
CB1002A-B.ADA	P
CB1003A-AB.ADA	P
CB1004A-AB.ADA	P
BB2001A-AB.ADA	P
BB2002A-AB.ADA	P
BB2003A-AB.ADA	P
BB2003B-AB.ADA	P
BB2003C-AB.ADA	P
CB2004A-B.ADA	P
CB2005A-B.ADA	P
CB2006A-AB.ADA	P
CB2007A-AB.ADA	P
BB3001A-B.ADA	P
BB3002A-AB.ADA	P
CB3003A-B.ADA	P
CB3004A-AB.ADA	P
BB3005A-AB.ADA	P
CB4001A-AB.ADA	P
CB4002A-AB.ADA	P
CB4003A-AB.ADA	P
CB4004A-B.ADA	P
CB4005A-AB.ADA	P
CB4006A-B.ADA	P
CB4007A-AB.ADA	P
CB4009A-AB.ADA	P
CB5001A-B.ADA	P
CB5001B-B.ADA	P

BC1001A-B.ADA	P	BC1303D-AB.ADA	P
BC1002A-B.ADA	P	BC1303E-AB.ADA	P
CC1004A-AB.ADA	P	CC1304A-AB.ADA	P
BC1005A-AB.ADA	P	CC1305B-AB.ADA	P
BC1006B-AB.ADA	P	BC1306A-B.ADA	P
BC1006C-AB.ADA	P	CC1307A-AB.ADA	P
BC1009A-AB.ADA	P	CC1308A-AB.ADA	P
CC1010A-AB.ADA	P	CC1310A-AB.ADA	P
CC1010B-AB.ADA	P	BC13A6A-B.ADA	P
BC1011A-AB.ADA	P	BC20C1B-AB.ADA	P
BC1011B-AB.ADA	P	BC2001C-AB.ADA	P
BC1012A-AB.ADA	P	CC2002A-AB.ADA	P
BC1013A-B.ADA	W	BC20A8A-B.ADA	P
BC1014A-B.ADA	P	BC3002A-AB.ADA	P
BC1014B-B.ADA	P	BC3002B-AB.ADA	P
BC1014C-B.ADA	P	BC3002C-AB.ADA	P
BC1014D-B.ADA	P	BC3002D-AB.ADA	P
BC1014E-B.ADA	P	BC3002E-AB.ADA	P
BC1014F-B.ADA	P	BC3003A-AB.ADA	P
BC1014G-B.ADA	P	BC3003B-AB.ADA	P
BC1014H-B.ADA	P	CC3004A-B.ADA	P
BC1014I-B.ADA	P	BC3005A-AB.ADA	P
BC1014J-B.ADA	P	BC3006A-AB.ADA	P
BC1101A-AB.ADA	P	CC3007A-AB.ADA	P
BC1102A-B.ADA	P	BC3009A-B.ADA	P
BC1103A-B.ADA	P	BC3009B-B.ADA	P
BC1104A-B.ADA	P	BC3009C-B.ADA	P
BC1104B-B.ADA	P	CC3011A-B.ADA	P
BC1106A-AB.ADA	P	BC3011B-B.ADA	P
BC1107A-B.ADA	P	BC3011C-AB.ADA	P
BC1108A-B.ADA	P	CC3011D-B.ADA	P
BC1109A-B.ADA	P	CC3012A-AB.ADA	P
BC1109C-AB.ADA	P	BC3013A-AB.ADA	P
BC1109D-AB.ADA	P	BC3014A-B.ADA	P
BC1109E-AB.ADA	P	BC3015A-B.ADA	P
BC1110A-B.ADA	P	BC3015C-B.ADA	P
BC1110B-AB.ADA	P	BC3015D-B.ADA	P
BC1110C-AB.ADA	P	BC3016A-B.ADA	P
BC1110D-AB.ADA	P	BC3016B-B.ADA	P
BC1110E-AB.ADA	P	BC3016C-B.ADA	P
BC1110F-AB.ADA	P	BC3016D-B.ADA	P
BC1110G-AB.ADA	P	BC3016E-B.ADA	P
CC1120A-B.ADA	P	BC3016F-B.ADA	P
BC1122A-B.ADA	P	BC3016G-B.ADA	P
BC1122B-B.ADA	P	CC3120A-AB.ADA	P
BC1202A-B.ADA	P	CC3120B-B.ADA	P
BC1202B-B.ADA	P	CC3125A-B.ADA	P
BC12A6A-B.ADA	P	BC31A6A-B.ADA	P
BC12ACA-B.ADA	P	BC31ACA-B.ADA	P
BC12ACB-B.ADA	P	BC31ADA-B.ADA	P
CC1301A-B.ADA	P	BC3201A-B.ADA	P
CC1302A-AB.ADA	P	BC3201B-AB.ADA	P
BC1303A-AB.ADA	P	BC3201C-B.ADA	P
BC1303B-AB.ADA	P	BC3202A-B.ADA	P
BC1303C-AB.ADA	P	BC3202B-B.ADA	P

BC3204C-B.ADA	P	BC3405E-AB.ADA	P
CC3203A-B.ADA	P	BC3405F-AB.ADA	P
BC3203B-B.ADA	P	CC3406A-AB.ADA	P
BC3204A-B.ADA	W	CC3406B-AB.ADA	P
BC3204B-B.ADA	W	CC3406C-AB.ADA	P
BC3204C0-B.ADA	W	CC3406D-B.ADA	P
BC3204C1M-B.ADA	W	CC3407A-AB.ADA	P
BC3204C2-B.ADA	W	CC3407B-AB.ADA	P
BC3204D-B.ADA	W	CC3407C-AB.ADA	P
BC3204E-B.ADA	P	CC3407D-AB.ADA	P
BC3205A-B.ADA	W	CC3407E-AB.ADA	P
BC3205B-B.ADA	W	CC3407F-AB.ADA	P
BC3205C-B.ADA	W	CC3408A-AB.ADA	P
BC3205D0-B.ADA	W	CC3408B-AB.ADA	P
BC3205D1M-B.ADA	W	CC3408C-AB.ADA	P
BC3205D2-B.ADA	W	CC3408D-B.ADA	P
BC3205E-B.ADA	P	BC3501A-AB.ADA	P
BC3205F-B.ADA	P	BC3501B-AB.ADA	P
CC3206A-AB.ADA	P	BC3501C-AB.ADA	P
CC3206B-AB.ADA	P	BC3501D-AB.ADA	P
BC3207B-B.ADA	W	BC3501E-AB.ADA	P
BC3208A-B.ADA	P	BC3501F-AB.ADA	P
BC3208A-B.ADA	P	BC3501G-AB.ADA	P
BC3301A-AB.ADA	P	BC3501H-AB.ADA	P
BC3301B-AB.ADA	P	BC3501I-AB.ADA	P
BC3302A-AB.ADA	P	BC3501J-AB.ADA	P
BC3302B-AB.ADA	P	BC3501K-AB.ADA	P
BC3303A-AB.ADA	P	BC3502A-AB.ADA	P
BC3304A-AB.ADA	P	BC3502B-AB.ADA	P
CC3305A-AB.ADA	P	BC3502C-AB.ADA	P
CC3305B-AB.ADA	P	BC3502D-AB.ADA	P
CC3306C-AB.ADA	P	BC3502E-AB.ADA	P
CC3306D-AB.ADA	P	BC3502F-AB.ADA	P
BC3307A-B.ADA	P	BC3502G-AB.ADA	P
BC3307A-B.ADA	P	BC3502H-AB.ADA	P
BC3308A-B.ADA	P	BC3502I-AB.ADA	P
BC3308A-B.ADA	P	BC3502J-AB.ADA	P
BC3401A-AB.ADA	P	BC3502K-AB.ADA	P
BC3401B-AB.ADA	P	BC3502L-AB.ADA	P
BC3402A-AB.ADA	P	BC3502M-AB.ADA	P
BC3402B-AB.ADA	P	BC3502N-AB.ADA	P
BC3403A-AB.ADA	P	BC3502O-AB.ADA	P
BC3403B-AB.ADA	P	BC3503A-B.ADA	W
BC3403C-AB.ADA	P	BC3503B-B.ADA	P
BC3404A-AB.ADA	P	BC3503C-B.ADA	P
BC3404B-B.ADA	P	BC3503D-B.ADA	P
BC3404C-AB.ADA	P	BC3503F-B.ADA	P
BC3404D-AB.ADA	P	CC3504A-B.ADA	P
BC3404E-AB.ADA	P	CC3504B-B.ADA	P
BC3404F-AB.ADA	P	CC3504C-B.ADA	P
BC3405A-AB.ADA	P	CC3504D-B.ADA	P
BC3405B-B.ADA	W	CC3504E-B.ADA	P
BC3405D-AB.ADA	P	CC3504F-B.ADA	P

ACVC 1.5 test results chapter 12 IABG - AVF

CC3504G-B.ADA P
CC3504H-B.ADA P
CC3504I-B.ADA P
CC3504J-B.ADA P
CC3504K-B.ADA P
CC3601C-AB.ADA P
CC3602A-AB.ADA P

ACVC 1.0 test results chapter 14 IABG - AVF

AE2101A-B.ADA	P	CE2401E-B.ADA	P
AE2101B-B.ADA	P	CE2401F-B.ADA	P
AE2101C-B.DEP	P	CE2402A-B.ADA	P
AE2101D-B.ADA	P	CE2404A-B.ADA	P
BE2101E-B.ADA	P	CE2405B-B.ADA	P
CE2102A-B.ADA	P	CE2406A-B.ADA	P
CE2102B-B.ADA	P	CE2407A-B.ADA	P
CE2102C-B.TST	P	CE2408A-B.ADA	P
CE2102D-B.ADA	N/A	CE2409A-B.ADA	P
CE2102E-B.ADA	N/A	CE2410A-B.ADA	P
CE2102F-B.ADA	N/A	BE3001A-B.ADA	P
CE2102G-B.ADA	N/A	BE3002A-B.ADA	P
CE2103A-B.TST	P	CE3002B-B.TST	P
CE2103B-B.TST	P	CE3002C-B.TST	P
CE2104A-B.ADA	P	CE3002D-B.ADA	P
CE2104B-B.ADA	P	BE3002E-B.ADA	P
CE2105A-B.ADA	P	CE3002F-B.ADA	P
CE2105A-B.ADA	N/A	AE3101A-B.ADA	P
CE2107A-B.ADA	P	CE3102A-B.ADA	P
CE2107B-B.ADA	N/A	CE3102B-B.TST	P
CE2107C-B.ADA	N/A	BE3102C-B.ADA	P
CE2107D-B.ADA	N/A	CE3103A-B.ADA	P
CE2107E-B.ADA	P	CE3104A-B.ADA	P
CE2108A-B.ADA	N/A	BE3105A-B.ADA	P
CE2108B-B.ADA	P	CE3107A-B.TST	P
CE2108C-B.ADA	N/A	CE3108A-B.ADA	P
CE2108D-B.ADA	P	CE3108B-B.ADA	P
CE2109A-B.ADA	P	CE3109A-B.ADA	P
CE2110A-B.ADA	P	CE3110A-B.ADA	P
CE2110B-B.ADA	N/A	CE3111A-B.ADA	P
CE2111A-B.ADA	P	CE3111B-B.ADA	N/A
CE2111B-B.ADA	P	CE3111C-B.ADA	N/A
CE2111C-B.ADA	P	CE3111D-B.ADA	N/A
CE2111D-B.ADA	N/A	CE3111E-B.ADA	N/A
BE2111A-B.ADA	P	CE3112A-B.ADA	N/A
BE2112B-B.ADA	P	CE3112B-B.ADA	P
BE2112C-B.ADA	P	CE3114A-B.ADA	P
BE2114A-B.ADA	P	CE3114B-B.ADA	N/A
CE2201A-B.ADA	P	CE3115A-B.ADA	N/A
CE2201B-B.ADA	P	CE3201A-B.ADA	P
CE2201C-B.ADA	P	CE3202A-B.ADA	P
CE2201D-B.DEP	P	CE3203A-B.ADA	P
CE2201E-B.DEP	P	BE3203A-B.ADA	P
CE2201F-B.ADA	P	CE3204A-B.ADA	P
CE2201A-B.ADA	P	CE3205A-B.ADA	P
CE2204A-B.ADA	P	CE3301A-B.ADA	P
CE2204B-B.ADA	P	CE3301B-B.ADA	P
BE2205A-B.ADA	P	CE3301C-B.ADA	P
CE2210A-B.ADA	P	CE3302A-B.ADA	P
CE2401A-B.ADA	P	CE3303A-B.ADA	P
CE2401B-B.ADA	P	CE3305A-B.ADA	P
CE2401C-B.ADA	P	CE3402A-B.ADA	P
CE2401D-B.DEP	P	CE3403B-B.ADA	P

ACVC 1.0 test results chapter 14 IABG - AVF

CE3402C-B.ADA	P	CE3605B-B.ADA	P
CE3402D-B.ADA	P	CE3605C-B.ADA	P
CE3402E-B.ADA	P	CE3605D-B.ADA	P
CE3403A-B.ADA	P	CE3605E-B.ADA	P
CE3403B-B.ADA	P	CE3606A-B.ADA	P
CE3403C-B.ADA	P	CE3606B-B.ADA	P
CE3403D-B.ADA	P	CE3606C-B.ADA	P
CE3403E-B.ADA	P	CE3701A-B.ADA	P
CE3403F-B.ADA	P	CE3702A-B.ADA	P
CE3404A-B.ADA	P	CE3703A-B.ADA	P
CE3404B-B.ADA	P	CE3704A-B.ADA	P
CE3404C-B.ADA	P	CE3704B-B.ADA	P
CE3405A-B.ADA	P	CE3704C-B.ADA	P
CE3405B-B.ADA	P	CE3704D-B.ADA	P
CE3405C-B.ADA	P	CE3704E-B.ADA	P
CE3405D-B.ADA	P	CE3704F-B.ADA	P
CE3406A-B.ADA	P	CE3704M-B.ADA	W
CE3406B-B.ADA	P	CE3704N-B.ADA	P
CE3406C-B.ADA	P	CE3705A-B.ADA	P
CE3406D-B.ADA	P	CE3706C-B.ADA	P
CE3407A-B.ADA	P	CE3706D-B.ADA	P
CE3407B-B.ADA	P	CE3706F-B.ADA	P
CE3407C-B.ADA	P	CE3706G-B.ADA	P
CE3408A-B.ADA	P	CE3707A-B.ADA	P
CE3408B-B.ADA	P	CE3708A-B.ADA	P
CE3408C-B.ADA	P	CE3709A-B.ADA	P
CE3409A-B.ADA	P	CE3801A-B.ADA	P
CE3409B-B.ADA	P	CE3802A-B.ADA	P
CE3409C-B.ADA	P	CE3803A-B.ADA	P
CE3409D-B.ADA	P	CE3804A-B.ADA	P
CE3409E-B.ADA	P	CE3804B-B.ADA	P
CE3409F-B.ADA	P	CE3804C-B.ADA	P
CE3410A-B.ADA	P	CE3804D-B.ADA	P
CE3410B-B.ADA	P	CE3804E-B.ADA	P
CE3410C-B.ADA	P	CE3804F-B.ADA	P
CE3410D-B.ADA	P	CE3804G-B.ADA	P
CE3410E-B.ADA	P	CE3804I-B.ADA	P
CE3410F-B.ADA	P	CE3804K-B.ADA	P
CE3411A-B.ADA	P	CE3804M-B.ADA	P
CE3411C-B.ADA	P	CE3805A-B.ADA	P
CE3412A-B.ADA	P	CE3805B-B.ADA	P
CE3412C-B.ADA	P	CE3806A-B.ADA	P
CE3413A-B.ADA	P	CE3806C-B.ADA	P
CE3413C-B.ADA	P	CE3806D-B.ADA	P
CE3501A-B.ADA	P	CE3806E-B.ADA	P
CE3601A-B.ADA	P	CE3806A-B.ADA	P
CE3602A-B.ADA	P	CE3809A-B.ADA	P
CE3602B-B.ADA	P	CE3810A-B.ADA	P
CE3602C-B.ADA	P	CE3901A-B.ADA	P
CE3602D-B.ADA	P	CE3902A-B.ADA	P
CE3603A-B.ADA	W	CE3903A-B.ADA	P
CE3604A-B.ADA	W	CE3905A-B.ADA	P
CE3605A-B.ADA	P	CE3905B-B.ADA	P

ACVC 1.5 test results chapter 14 IMAG - AVF

CE3905C-B.ADA P
CE3905L-B.ADA P
CE3906A-B.ADA P
CE3906B-B.ADA P
CE3906C-B.ADA P
CE3906D-B.ADA P
CE3906E-B.ADA P
CE3906F-B.ADA P
CE3907A-B.ADA P
CE3908A-B.ADA P

APPENDIX E

VERSION 1.6 WITHDRAWN TESTS

The following tests have been withdrawn from Version 1.6 of the Ada Compiler Validation Capability (ACVC) for the reasons given below.

- C35904A-B: The elaborations of the subtype declarations for SFX3 and SFX4 in this test raise NUMERIC_ERROR in some implementations. The exception is raised on the conversion of the real literals 2.0 and 5.0 to the base type of FIX.

- B38105B-AB: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- C45521[6]-B: Cases C and I define the model interval for the result too narrowly.

- C48005C-B: Lines 38 and 63 of this test should check that the value of the designated object is null.

- C48006B-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- C64103C-B: This test should raise CONSTRAINT_ERROR during the conversion at line 179.

- C64103D-B: This test involves a CONSTRAINT_ERROR vs. NUMERIC_ERROR issue that is to be resolved by the Language Maintenance Committee.

- C64105E-AB: For case E, ensure that non-null dimensions of formal and actual parameters belong to both index subtypes (see AI-00313).

- C64105F-AB: For case E, ensure that non-null dimensions of formal and actual parameters belong to both index subtypes (see AI-00313).

- B66001A-B: This test checks (in section G) that a function without parameters, which is equivalent to an enumeration literal in the same declarative region, is a redeclaration and as such is forbidden. According to RM 8.3(17), the explicit declaration of such a function is allowed if an enumeration literal is considered to be an implicitly declared predefined operation. The RM is not clear on this point. This issue has been referred to the Language Maintenance Committee for resolution. Since the issue cannot be resolved at this time, the test is withdrawn from Version 1.6.

- B67001A-B: Line 414 is missing the "BEGIN NULL; END;" needed to complete the block beginning at line 389 (case H).

- B67004A-B: The default name for a formal generic equality function should not be allowed to be "/=" unless an expanded name is used.

- B74103F-B: This test hinges on whether or not a generic formal type declaration declares a type. This matter will be debated by the Language Maintenance Committee in November.

- B74207A-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- C93005A-B, C93005B-B, C93005C-B: These tests contain a declaration of an integer variable whose initialization is solely for the purpose of raising an exception. Some compilers will not raise this exception due to their optimization.

- C93007B-B: This test should check for PROGRAM_ERROR rather than TASKING_ERROR (SEE AI-000149).

- CA1003B-AB: A compilation that contains an illegal compilation unit may now be rejected as a whole (see AI-00255/05).

- CA1011A -B: The test objective should be reversed to be consistent with AI-00199.

- CA1108A-B: A pragma ELABORATE is needed for OTHER_PKG at line 25.

- CA1108B-B: A pragma ELABORATE is needed for FIRST_PKG at line 39 and for LATER_PKG at line 49.

- BA2001E -AB: LRM 10.2(5) states that "simple names of all subunits that have the same ancestor library unit must be distinct identifiers." This test requires that the above conditions be checked when the stub is declared; but since the LRM uses the term "subunit", it is not clear that the check must be made then, as opposed to when the subunit is compiled. (There may be an LMC ruling regarding this issue.)

- CA2009B-B: The repetition of the main procedure after the subunit body makes the subunit body obsolete; therefore, an attempt to execute the main procedure will fail.

- CA2009E-B: The repetition of the main procedure after the subunit body makes the subunit body obsolete; therefore, an attempt to execute the main procedure will fail.

- CA2009F -B: The file CA2009F2-B is missing from this test suite.

- BC1013A-B: The declaration of equality in lines 86-87 is illegal because the parameter type T declared in line 11 is not a limited type (LRM 6.7-4).

- BC3204A-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3204B-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3204C -B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3204D-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3205A-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3205B-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3205C-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3205D -B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3220B-B: This test assumes that instantiated types may be static. This assumption has been questioned, and the matter will be considered by the LMC.

- BC3405B-B: Instantiations with types that have default discriminants are now legal (see AI-00037).

- BC3503A-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- CE2107E-B: This test has a variable, TEMP_HAS_NAME, that needs to be given an initial value of TRUE.

- CE3603A-B: The last case is inconsistent with AI-00050. If string argument is null, no attempt to read is made and END_ERROR is not raised.

- CE3604A-B: Cases 5, 8, 9, and 11 are inconsistent with AI-00050. SKIP_LINE is called only if the end of the output string has not been met.

- CE3704M-B: A superfluous SKIP_LINE causes the input and output operations to be out of synchronization.

END

DTIC

6-86